Shah Reezal Bin Ahmad
WET 000223

Interactive 3D Human Head Multimedia System

Under the supervision of
Pn. Nurul Fazmidar Mohd. Noor
And moderated by
Dr. Phang Keat Keong

Abstract

This document was organized in order to give to the reader or user a complete overview about the upcoming project, in the sense that the user should learn the theoretical concepts first and then be aware of how the concepts be applied, being presented meanwhile, some introduction, literation research, methodology, system analysis, system design, system development, system testing, the discussion on the problems occur and examples related with "Interactive 3D human Head Multimedia System".

My advisor and supervisor for this thesis is Pn. Nurul Fazmidar Mohd. Noor and the moderator is Dr. Phang Keat Keong. The topic I chose is called "Interactive 3D Human Head Multimedia System". It's a web based project and mostly for the purpose of educational, training, medicine and extra general knowledge. I found out that the most important thing in developing a system is that it must be user friendly, easy-to-use, reliable, trustworthy, efficient and fast interaction. Prototyping is important to obtain comments and suggestions from the users to ensure that the final product is high in quality and meet the user's requirements and expectations.

This project is concerned with the signal-to-symbol problems of building skinned, segmented, land marked and labeled 3D models of the human head from range data. A fully automated model-based process is presented that takes raw range data, clean and skin it, and then locates "interesting" features, to enrich the surface with symbolic information for specific applications. I obtain a lot of useful experience and knowledge through this project undertaking.

Acknowledgement

One of my greatest pleasures of writing this report is acknowledging the efforts of many people ranging from my academics advisors, lecturers to my friends. I would like to acknowledge the great efforts put by the Faculty of Science Computer & Information Technology staffs that provided valuable suggestions and encouragement. They are Pn. Nurul Fazmidar Mohd. Noor, Dr. Phang Keat Keong, Ms. Nor Aniza Abdullah and all the lecturers who helped me through with their wisdom and knowledge. I would like to thank Pn. Nurul Fazmidar Mohd. Noor for her advices, support and time spend on discussion; to Dr. Phang Keat Keong, for his encouragement and confidence. From my point of view, both my supervisor and moderator are open-minded people, easy in taking new suggestions and I enjoyed every minute working to finish this report with them.

I'm truly amazed by their in-depth knowledge of computer science and information technology. I sincerely appreciate their comments and suggestions. Thanks also to all my friends whose comments have helped shape my report, especially Nik Ezami Nik Ismail, Tay Kiang Heng, Mohd. Irfan Abadi and Umi Aimah Abu Bakar. All their names may not appear on this report but whose cooperation, guidance and understanding were crucial to the success of my report.

Finally, I'd like to thank my family for all the supports they have given me.

Content

Abstract ii

Acknowledgement iii

Content iv

List of Figures viii

Chapter 1 Introduction 1

1.1 Introduction 2

1.2 Interactive 3D Human Head Multimedia System 2

1.3 Project Objectives 4

1.4 Project Scopes 6

1.5 Project Development Plan 8

Chapter 2 Literature Review 13

2.1 Literature Review 14

2.1.1 Caput V 2.02 SE 15

2.1.2 Visible Productions 17

2.1.3 Sectional Anatomy of the Human Head 21

2.2 System Drawbacks 22

2.3 Comparison 24

2.4 Questionnaire 25

Chapter 3 Methodology 31

3.1 Software Process Model 32

3.2 The Waterfall Model 32

3.3 The Waterfall Model with Prototyping 34

3.4 System Development Lifecycle 42

3.5 The Chosen Methodology 49

Chapter 4 System Analysis 50

4.1 System Analysis 51

4.2 Functional requirements 52

4.3 Nonfunctional requirements 53

4.4 Software Requirements 55

4.4.1 Virtual Reality Modeling Language (VRML 2.0) 55

4.4.2 Cosmoplayer 66

4.4.4 Macromedia DreamWeaver MX 77

4.4.5 Adobe Photoshop 7.0 78

4.4.6 Macromedia Flash MX 79

4.4.7 Sound Forge Studio 6.0 80

4.5 Hardware Requirements 81

Chapter 5 System Design 82

5.1 Conceptual Design / System Design 83

5.2 Representation of Muscle and Dividing of Model 87

5.3 The Surface Anatomy and Surface Markings 88

5.4 Screens and Interfaces Design 95

5.5 Storyboard 96

Chapter 6 System Development & Implementation 97

6.1 Introduction 98

6.2 Development Environment 98

6.2.1 Development Tools 99

6.2.1.1 Hardware Requirements 99

6.2.1.2 Software Requirements 99

6.2.1.3 Methodology 100

6.2.2 System Coding 101

6.2.2.1 Coding Approach 101

6.2.2.2 Coding Style 102

6.3 Project Development 103

6.3.1 New Software Requirements 104

6.3.2 Data Preparation 104

6.4 Summary 105

6.5 The User Manual 106

Chapter 7 System Testing 115

7.1 Introduction 116

7.2 Testing Objectives 117
7.3 Testing Technique 117
7.3.1 White Box Testing 117
7.3.2 Black Box Testing 118
7.4 Testing Strategy 119
7.4.1 Unit Testing 119
7.4.2 Unit Testing Example 120
7.4.3 Control Object Testing 122
7.4.4 Integration Testing 122
7.4.5 System testing 123
7.4 Summary 123
Chapter 8 System Evaluation 124
8.1 Introduction 125
8.2 Problems Encountered And Solutions 125
8.3 System Strengths 126
8.4 System Constraints 129
8.5 Future Enhancements 129
8.6 Knowledge and Experience Gained 130
8.7 Summary 132
Conclusion 133
Appendix 134
Reference 139
Bibliography 140

1.1 Introduction

Modern multimedia computer systems are able to present rich and realistic information to the users. Advanced technology of graphic hardware enables common use of the 3-dimensional graphics. There are possibilities to create systems that use virtual humans and animate them in real time. Virtual human can be represented either by the whole body or only by head or face. The virtual human faces are often able to depict expression, even though these systems usually depict human speech. Such virtual humans have a large field of use. They can be part of web pages, educational programs, language training, animated agents etc. Virtual humans are unusual way to present information to the handicapped people. For example deaf people can practice their ability to read lips of other person. The creation of a virtual human face is complex task. We have to find a proper method for data representation of the model. The next step is a creation of the facial muscles and subdividing of the model.

1.2 Interactive 3D Human Head Multimedia System

The project that was given to me is a web-based 3D modeling of a human head multimedia system. The use of VRML (Virtual Reality Modeling Language) coding is essential for the success of this project. Model representation with proper data representation is necessary for an easy modification of existing models and for creating of the new models of virtual humans. Model of a face could be represented in the several

ways. The developers of virtual human systems often create their own data formats or they use existing formats such as VRML.

i. **Raw data**

This type of representation usually uses one or more data files. The files usually contain a set of numbers. These numbers describe vertices, polygons (usually triangles), colors and control points or vectors of the model. A few data files are used in the application of Expression toolkit. These files contain coordinates of vertices, indexes of polygon vertices and coordinates of vectors.

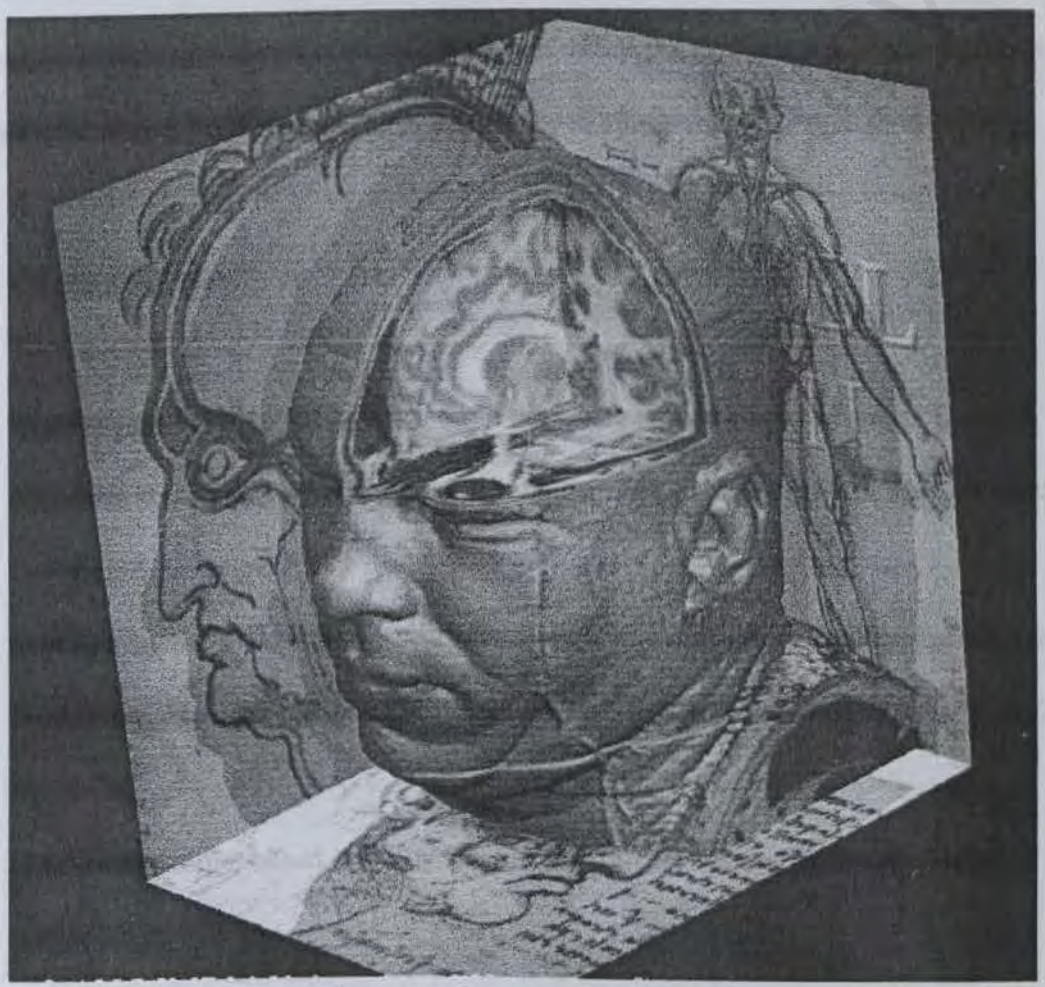


Figure 1.1.1 3D Human Head

5) Each different level of intellects can gain knowledge about the human head anatomy through this 3D model which is designed to be user-friendly.

6) Users and students can explore the depth of the human head in the comfort of their own homes because it is compact in the form of CD. Which it is easily carry around and light weight.

7) Provide a simple and user-friendly system to obtain more information about the human head.

8) Designed to the needs of educational purposes, thesis's, school projects, research and as references.

9) Presents a human head modeling system that can define and manipulate the geometry of a human face in real-time.

10) In addition to more general VRML concepts a number of projects present possible virtual solutions to complex spatial and interaction scenarios encountered in some of the research themes within the studio.

Figure 1.3.1 Human Head Geometry

1.4 Project Scopes

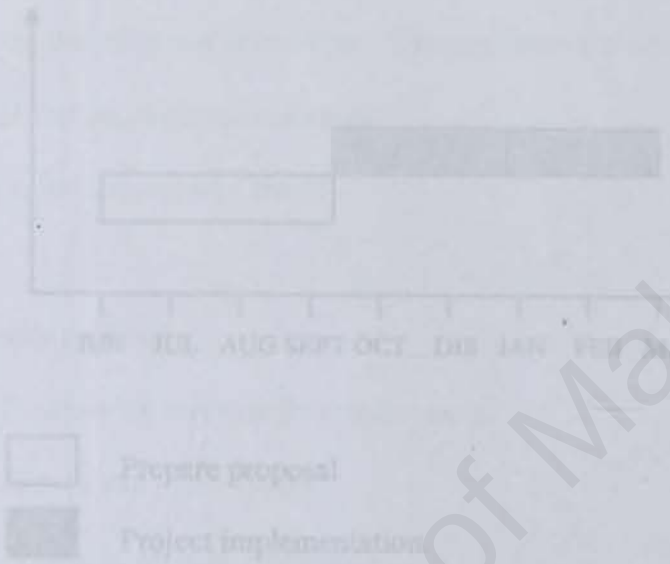
As the general view of this website content, I would say that it's covered as low as what high school students learn in their biology class or maybe primary school students for their own extra knowledge, who knows, and up to what medical students learn in their anatomy subject. And not to forget the doctors who'll be fascinate by this website when they surf for extra information because of the human head details, the interfaces, the graphics, color combination and so much more!!

This website provides an attractive 3D model that can be navigated and it will show the nearest actual looks of a human head. It provides the human head description and function. And also a trial session for users to see how the human head and it's anatomy inside and outside functions.

The target users for this website are students of all level of education, pre-university, medical students, doctors and public users of all age. As a school student or pre-university students, they only learn the human head anatomy for schooling purposes or school projects and some of them have no intention to learn more that what have written in their syllabus. So in this case, I would like to say that my main target users are the medical students or doctors. They have to get a clear view on this system because soon they will apply this knowledge to an actual human being.

Public users learn about this topic because they feel that it is essential for people to know how their body works, it is like a general knowledge for them. While doctors are surfing to recall back what they have learn in their schooling time through this website.

Let's focus the main target user of this project to the students and for those seeking knowledge on such field. The availability of this website is the only constraint to archive these valuable contents. This is because this website could only be reached through internet access.



1.5 Project Development Plan

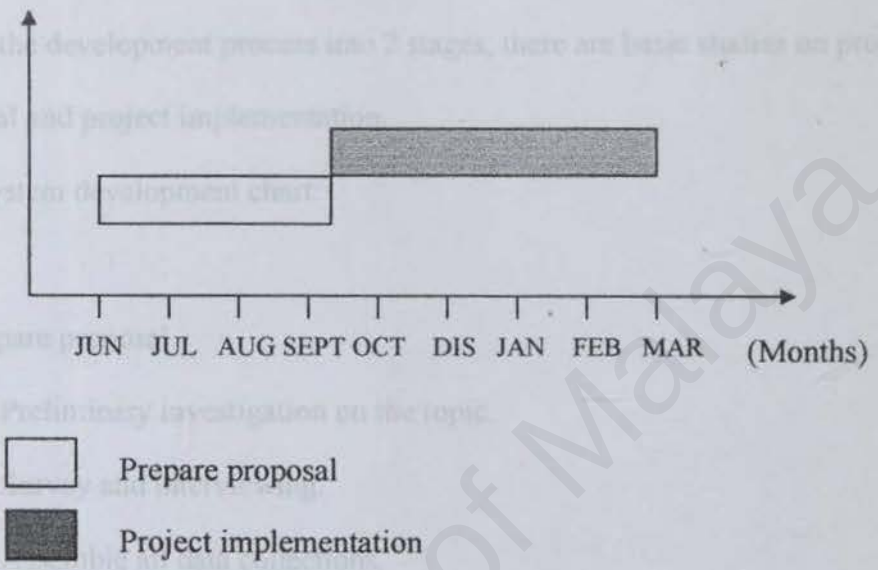


Figure 1.5.1 System development Chart

System Development Chart

As the times to develop this project are limited, I have to carefully estimate the period for every development stages. This step is basically crucial because with only slight mistake it will ruin the whole development process.

I have divided the development process into 2 stages; there are basic studies on project, writing proposal and project implementation.

Below is the system development chart:

a) Prepare proposal

1. Preliminary investigation on the topic.
2. Survey and interviewing.
3. Assemble all data collections.
4. Specify system requirements.
5. Specify system design.
6. Writing proposal.

b) System Implementation

1. Assemble all data.
   - Assemble all the specific data about the human head.
   - List out all the image and 3D models that have to build.
2. Building 3D model part by part of a human head.
3. Integrate all the models to form fully 3D human head.
4. Create interfaces.

- Create virtual reality interfaces.
- Import 3D models into VRML.
- Building connections between pages.

5. Testing.
   - Errors detection.
   - Fixing the errors.
6. Bring the system online.

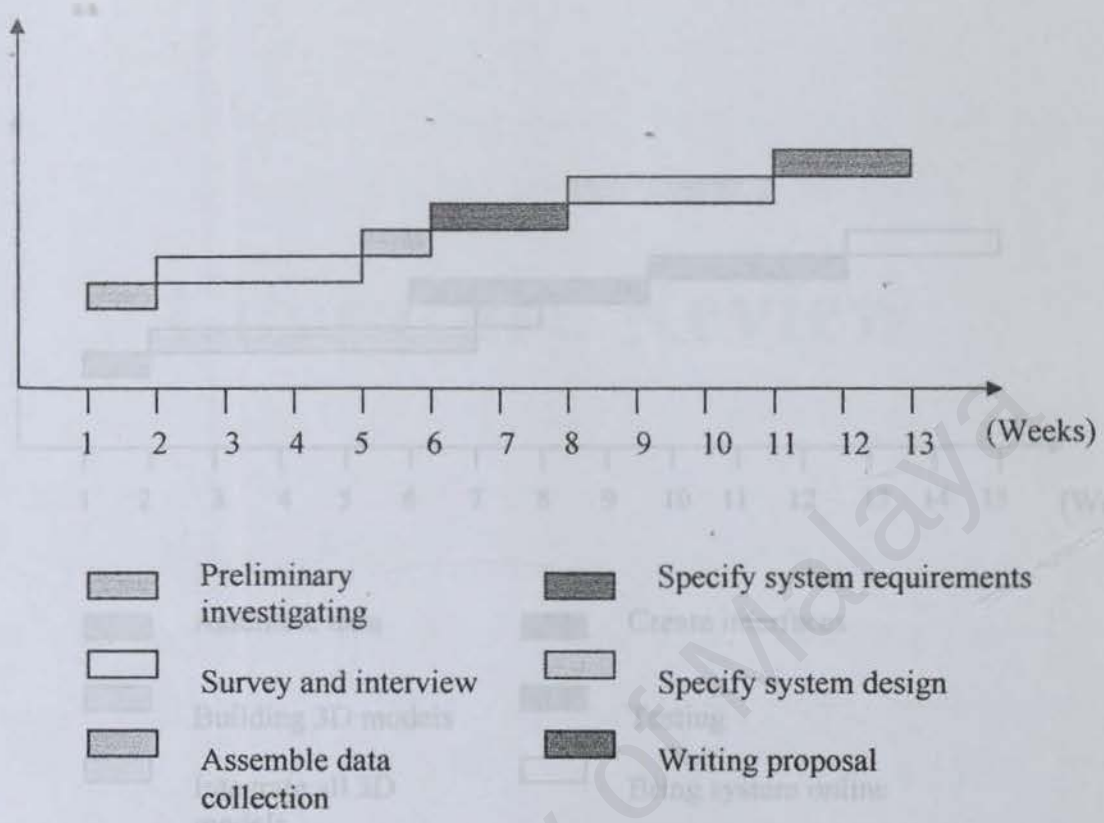


Figure 1.5.2 Proposal Timeline

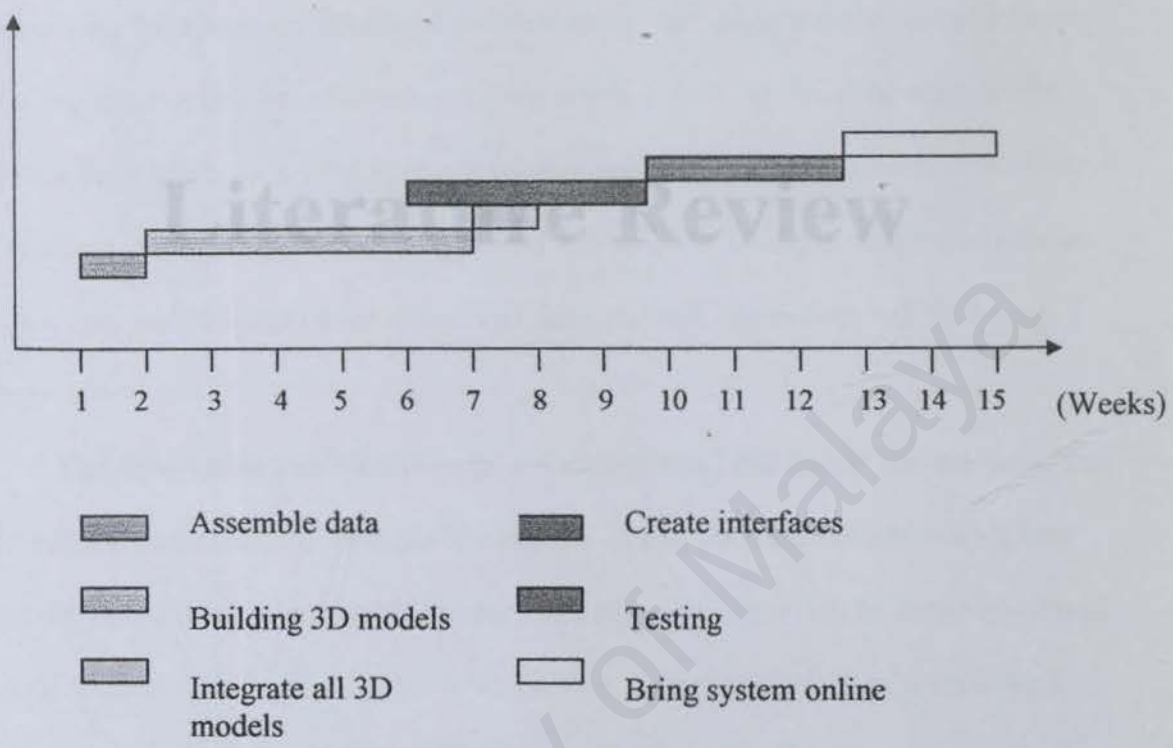


Figure 1.5.3 System Implementation Chart

Literature Review

2.1 Literature Review

This is the part where or I may say the chapter for me to compare the system that I tend to develop with the systems that have been developed earlier. I came out with these 3 quite intriguing systems that have been already developed, I shall list down the name of the websites, their specialties, what kind of output or product that they produce, date of last updated and many more. This is necessary for this project-making to succeed in its class. To gain trust and confident from clients and users. So that, our system will be the top system.

The comparison I've done towards the system that I can find in the internet is that I can see and do research to see what it is capable of and what kind of advantages they given. By doing this, I can simplify the purposes that my system will be better and it will provide whatever needs the users have. To provide the right information's, up-to-date details and specification of a human head anatomy, trustworthy system and easy to use interfaces. After some period of website searching of a similar topic to mine, that is "the interactive 3D human head multimedia system", I came across quite a few website with the similar information. I can say that, there's a lot, maybe thousand of websites with interactive 3D human head multimedia system. But what I find here, are 3 websites, the most competitive system to what I may be building soon.

2.1.1 Caput V 2.02 SE

http://www.3d-head.com/

Caput V2.02 SE is an interactive 3D atlas of the head and neck region by A.R. Viddeleer, MD and H.K.P. Feirabend, PhD University of Leiden, The Netherlands. This is the homepage of Caput, an interactive 3D atlas of the head and neck region. Caput is based on a complete 3D MRI set of the head. This atlas has been created to explain the spatial relations of anatomical structures in the head to students and physicians. The MRI images were chosen to get students acquainted to this rapidly evolving imaging technique. The classic anatomical atlas provides a flat representation of a three-dimensional situation. Hence it is difficult to see how structures relate to each other in space. Also, the structures are shown from only a small amount of views. Most of the current digital atlases have the same shortcomings, while these also display a fixed number of pre-prepared images. The approach of Caput is to let the *user* determine which structures will be (partially) visible and from which directions. Also, a stereoscopic mode is provided to elucidate the three-dimensional relations. Primarily, Caput was developed for training of students and residents. The existing interactive atlases with functions comparable to Caput's are very expensive (US $ 2000!) and only work on expensive

workstations. Therefore our primary concern was to develop a system that works on a plain PC and is still affordable.

Caput v2.02 SE Features:

- A complete sagittal 3D MRI set of the head and neck.
- Multiple cutting planes through the MRI set.
- A database of more than 400 anatomical structures.
- Very comprehensive en user-friendly search functions.
- Structures that can be selected in all slices and directions.
- Possibility to view one single point from different directions.
- A very fast 3D reconstruction option for many structures.
- Easy-to-use user adjustable 3D reconstructions.
- Function to find corresponding MRI slices of points in 3D reconstructions.
- Easy and very intuitive navigation tools.
- A stereoscopic mode to see the 3D reconstructions with depth.
- And much more...

2.1.2 Visible Productions

http://visiblep.com/

Visible Productions, Inc., a Colorado corporation, was formed in January 1996 by the merger of Micron Biosystems, Inc. and BioGraphics, Inc. The merger combined the assets of Micron's five years of multimedia publishing and distribution development with the assets of BioGraphics eleven years of anatomically accurate, state-of-the-art visual imagery for medical and veterinary clients worldwide. The combined entity is Visible Productions.

Visible Productions has developed the world's most extensive library of 3D digital anatomy. This powerful resource takes learners beyond the confines of a flat printed page, into an interactive, three-dimensional environment where spatial relationships and events that happen over a period of time can be fully explored. Physiological processes from the molecular level to the entire body systems have been created as part of award winning custom biomedical animations, multimedia programs and print images for clients worldwide.

Visible Productions' competitive advantage is derived from three factors:

- The Visible Human Project from which the models were created
- Proprietary software developed specifically to create the models
- Unique ability to create award winning multimedia programs

VP's models of male and female anatomy were created from the cryosection images from the National Library of Medicine's "Visible Human Project".

Custom animation & multimedia services

VP offers a full range of program development services from project planning and program development (scripting) to final packaging.

VP's custom content is widely used to:

- Simulate surgical procedures
- Design Continuing Medical Education (CME) programs
- Demonstrate procedures and processes
- Introduce new products to physicians and consumers
- Examine or flythrough anatomical structures and organs
- Showcase new products or devices at conferences and trade shows
- Develop sales training tools
- Design advertising and promotional materials
- Prepare graphics for professional and educational presentations
- Design WEB site content

Animation Services

Visible Productions' in-house team of anatomists, animators, designers, programmers, and multimedia specialists offer a full range of development services from storyboard to completed product. Using Visible Productions' custom 3D animations, anatomical fly-through and interactive tours of the human body, difficult concepts are made easier to understand.

VP animations are unmatched in providing anatomical accuracy, detail, and aesthetics. Use our custom animation services to:

- Develop engaging teaching tools for students at all levels
- Create effective training programs for healthcare professionals
- Showcase new research or products at conferences and tradeshows
- Develop patient education or compliance programs
- Design comprehensive sales training materials
- Clarify medical procedures and processes
- Simulate surgical procedures
- Create effective advertising campaigns
- Prepare custom presentation graphics
- Develop clear, accurate medical-legal presentations
- Add dynamic content to television productions

Multimedia Features

Incorporate custom features into your animations including:

Virtual Reality Dissections -An invaluable tool for visualizing all important spatial relationships in anatomy, biomechanics, surgical procedures.

Animation - Visualizing scientific concepts that involve change over time requires animation. This may include tissue movement, development, pathological changes, or physiology. These changes can be on a gross level or the same treatment can be applied to the cellular level. Drug modes of action or drug interactions are but a few examples that can be included in an interactive format.

Photo Realistic Texture Mapping -Lifelike tissue color and texture are applied to our reality-based anatomical models.

Roll-Over Hot Spot Labeling - Enables "on the fly" labeling of structures and features in our Virtual Reality Environments. Visible Productions' ability to call up labels and text boxes gives users a true interactive textbook at their fingertips.

Virtual Reality Panoramas - Offers 360° object rotation for unparallel views of anatomical features.

Fly-Through -Travel through internal structures of the body to view anatomy, physiology, virtual surgery, fluid dynamics, etc.

2.1.3 Sectional Anatomy of the Human Head

These images provide views of the human head in three distinct planes of sectioning (sagittal, transverse, and coronal) and were used by Dr. Negulesco in his laboratory demonstrations. Use the images to review your understanding of the three-dimensional anatomical organization of the head. As in the laboratory these images are not labeled, and you are encouraged to consult Human Cross-Sectional Anatomy by Ellis et al. for reference. Labeled images of the head & neck can also be found in Dr. Negulesco's Genesis.

Sagittal

Transverse

Coronal

2.2 System Drawbacks

There is thousands of website based on the similar topic as my project,"Interactive 3D Human Head Multimedia System" but I chose this three for my literature review. The first two websites, I have to admit that it is quite a competitor to my future system on human head. But what I can summarize has already listed below. And I have to add that From all the websites I obtained, there is one common purpose between us and that is to educate people and provide knowledge on the human head anatomy in 3D form.

What I found out is that, how you actually capture the interest of the users is what important to the system and its purpose of existing. The systems have to look interesting or intriguing, it's like once the user surf into our website, and they will feel challenged, challenged by the system. The first two website is a little bit too complex for students of primary and secondary stage. And their interfaces are dull. When people come across this kind of situation, they will feel bored and lost interest.

Plus, the navigation sequence is confusing for people to understand. We need a easy-to-use and user-friendly system to maintain their visit and constant use of the system. As for the third website, this is an example of very limited information on the human head anatomy. The model of the human head cannot be rotate. The background color is very dull plus no many navigation buttons.

Below are the summaries of what I found out on the systems drawback. They are all in point form for better understanding.

- It's not suitable for every level of intellects.
- Boring and dull interfaces.
- Limited navigation buttons; such as there are no "Back" buttons and the users cannot "Stop" at any time they want.
- Confusing page-by-page directions/links.
- Diagrams & pictures are not well-labeled.
- System ordering page does not exist except for "How to contact us" page.
- Difficult structures remain difficult to understand.
- Limited information on the human head.
- First page refresh/download time is too long.
- 3D model that cannot navigate or rotate.
- Segmentation of the human head is too complex for all type of intellect.
- The systems are not user-friendly.

2.3 Comparison

The system that I want to develop can be engage as a-teaching tool for students at all levels. Create effective training programs for medical students. This is because my target users are from the education background such as students, trainees or college students. So it is best to make it clear on the scope of this project. As for my 3D human head model it can be navigate (360 degree). With an almost realistic features of the human head for better understanding and to avoid misinterpretation.

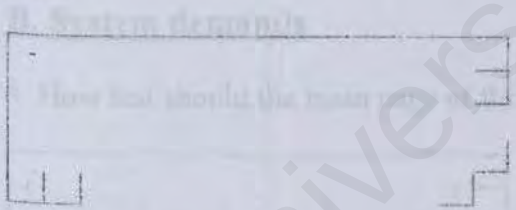
An option to choose information deliverable via human voice or display text. This is one of my extra features in the future system. Which will make the system more "alive" and interesting? Screen to display description or information text when the users click at a certain point on the human head. Plus, the user can chose by clicking the button that available, just how much information or details that they want to know. Demonstration for the human head functions by the system. So that it would make the user feel relax while surfing the website. Not causing them too much confusion or misunderstanding. One of my plans to achieve a user-friendly and easy-to-use system. Navigation buttons provided (with an extra buttons such as Back, Stop, Play and Print) Color combinations & graphics usage, which is important for the eyes of the users, to ensure it doesn't cause any irritation, such as bright color background or text fonts. Multi-channel users for different level of intellects.

A. General Information – Description of a virtual guidance system

1. What type of presentations do you believe should be supported by the system?

- [ ] 3D presentation (augmented to real objects-virtual reality system)
- [ ] 3D presentation (without augmentation-augmented reality system)
- [ ] Audio output
- [ ] Text documents (annotated to real objects)
- [ ] Text documents (without annotation)
- [ ] Video
- [ ] Images

Other:



2. How much improvement do you believe an augmented reality system could offer in an interactive 3D human head?

- (•) Not at all
- ( ) Enough
- ( ) Significant

Other

3. Do you need human voice for displaying information on the human head anatomy?

- Yes, because:

- No, because:

Comments

B. System demands

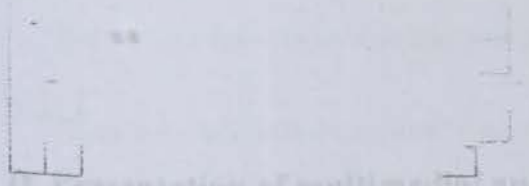
4. How fast should the main page of this website need to be fully downloaded?

5. In an augmented interactive 3D human head multimedia system it is possible to have in the site artificial objects (labels of dimensions 8cm x 8cm), which will help the system to identify visitor's position in the area or diagrams. Would you be disturbed by the existence of such objects?

- Yes

- No

Comments



6. How many of those artificial objects can exist in the area without disturbing you?

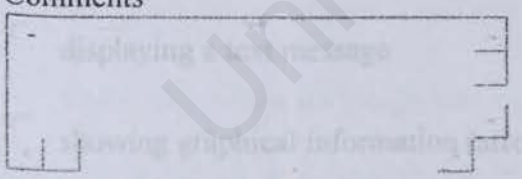
- Ten
- Five
- Two
- None

C. Multimedia Information Organizing

7. How much help would be a personalized guide during an augmented interactive 3D human head multimedia system?

- Not at all
- Enough
- Significant

Comments



8. What kind of information should the system use to build a personalized display?

- Time remaining
- Degree of detail

Other:

D. Presentation of multimedia/ audiovisual information

9. In an augmented interactive 3D human head multimedia system in what way would you prefer the tour to take place?

- Free tour in the site with the capability of requesting more information on a subject according to user's demands
- Directed/ Predefined tour and presentation of 3d and audiovisual information
- Directed tour with the capability of interfering during the presentation of multimedia information (repeat, pause, search etc.)

a) If your last answer was not a free tour, should the system indicate the way to the next interesting points or facts?

- Yes
- No

b) If Yes, how should the system do that?

- using a sound message
- displaying a text message
- showing graphical information (arrows, ...)
- using land marks

10. What options should the system provide during the tour?

- [ ] Help about the system services and the way it works
- [ ] More information concerning a specific subject
- [ ] Pause the flow of audiovisual information regarding an object or organ
- [ ] Repeat audiovisual information
- [ ] Choose an object and gain information about it (among those in your eyesight)

Other

11. What kind of display would you prefer?

- (•) Display texts
- ( ) Emergency sounds.

12. In 1-10 scale give the degree in which you would prefer to interact with the system using the following devices

- (•) Computer device (mouse, keyboard etc.)
- ( ) Voice recognition for simple commands
- ( ) 3D-pointing device

13. What information is important to receive from the system (please rank 1-10)?

- Original shape of the organ and facial features
- Change of the features over time
- Demonstration of specific function
- Visible Explanations
- Auditive Explanation
- Textual Explanation

Other

14. What level of visual granularity would be acceptable?

- I care to see only the shape of the human head
- I care to see advanced details of the human head
- I would like to be able to see every little detail of the human head

Through this questionnaire, I can estimate of how many people would like the system to function. Keep track between the users and my design. Asking about their (the users), their point of view on the system. Knowing if the system is over-design or not. Try to find out how the system functions in the eyes of the users.

Methodology

3.1 Software Process Model

As for the method that I chose to develop my system is the "Waterfall model with prototyping". But before I go on with the waterfall model with prototyping, let's see what goes on in the waterfall model first. One of the first models to be proposed is the "waterfall model", where the stages are depicted as cascading from one to another. As the figure implies, one development stage should be completed before the next begins. Thus, when all of the requirements are elicited from the customer, analyzed for completeness and consistency, and documented in a requirements document, then the development team can go on to system design activities. The waterfall model presents a very high level view of what goes on during development, and it suggests to developers the sequence of events they should expect to encounter.

3.2 The waterfall model

The waterfall model has been used to prescribe software development activities in a variety of contexts. Associated with each process activity were milestones and deliverables, so that project managers could use the model to gauge how close the project was to completion at a given time. For instance, "unit and integration testing" in the waterfall ends with the milestone "code modules written, tested and integrated;" The intermediate deliverable is a copy of the tested code. Next, the code can be turned over to the system testers so it can be merged with other system components (hardware or software) and tested as a larger whole. The waterfall model can be very useful in helping

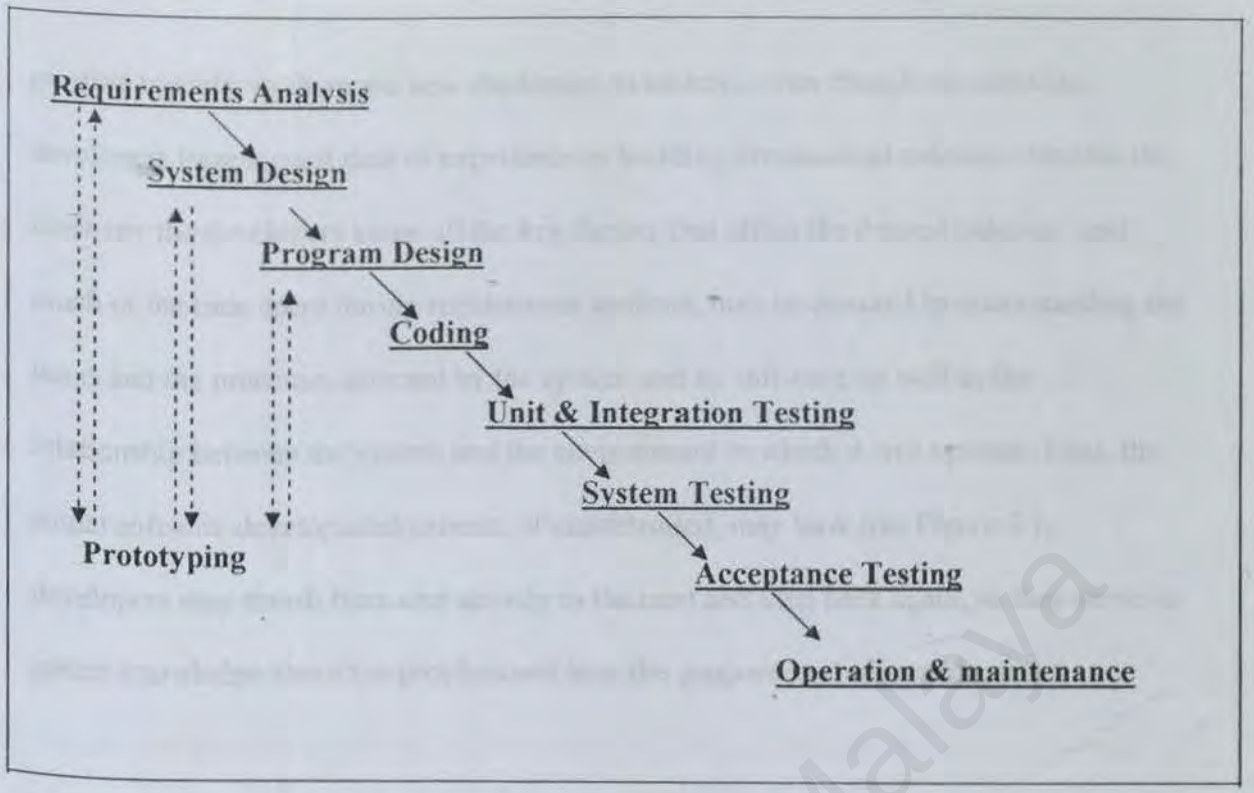


Figure 3.2.1 The waterfall model.

developers lay out what they need to do. Its simplicity makes it easy to explain to customers who are not familiar with software development; it makes explicit which intermediate products are necessary in order to begin the next stage of development. Many other, more complex models are really just embellishments of the waterfall, incorporating feedback loops and extra activities.

The biggest problem with the waterfall model is that it does not reflect the way code is really developed. Except for very well understood problems, software is usually developed with a great deal of iteration. Often, software is used in a solution to a problem that has never before been solved or whose solution must be upgraded to reflect some change in business climate or operating environment. For example, an airplane manufacturer may require software for a new airframe that will be bigger or faster than

existing models, so there are new challenges to address, even though the software developers have a great deal of experience in building aeronautical software. Neither the users nor the developers know all the key factors that affect the desired outcome, and much of the time spent during requirement analysis, may be devoted to understanding the items and the processes affected by the system and its software, as well as the relationship between the system and the environment in which it will operate. Thus, the actual software development process, if uncontrolled, may look like Figure 3.1, developers may thrash from one activity to the next and then back again, as they strive to gather knowledge about the problem and how the proposed solution addresses it.

3.3 The waterfall model with prototyping

The software development process can help to control the trashing by including activities and sub processes that enhance understanding. Prototyping is such a sub process; a prototype is a partially developed product that enables customers and developers to examine some aspect of the proposed system and decide if it is a suitable or appropriate for the finished product. For example, developers may build a system to implement a small portion of some key requirements to ensure that the requirements are consistent, feasible and practical; if not, revisions are made at the requirement stage, rather than at the more costly testing stage. Similarly, parts of the design may be prototyped, as shown in Figure 3.2. Design prototyping helps developers assess alternative design strategies and decide which is best for a particular project. The designers may address the requirements with several radically different designs to see

which has the best properties. For instance, a network may be built as a ring in one prototype and a star in another, and performance characteristics evaluated to see which structure is better at meeting performance goals or constraints.

Often, the user interface is built and tested as a prototype, so the users understand what the new system will be like, and the designers get a better sense of how the users like to interact with the system. Thus, major kinks in the requirements are addressed and fixed well before the requirements are officially validated during system testing; validation ensures that the system has implemented all of the requirements, so that each system function can be traced back to a particular requirement in the specification. System testing also verifies the requirements; verifications ensure that each functions works correctly. That is, validation makes sure that the developer is building the right product (according to the specification), and verification checks the quality of the implementation. Prototyping is useful for verification and validation.

3.3.1 Requirement analysis

When a customer requests that we build a system, the customer has some notion of what the system will do. In the Interactive 3D Human Head Multimedia System, the customers need to understand the ideas and the concept pf the developing project. No matter whether its functionality is old or new, each software-based system has a purpose, usually expressed in what the system can do. A requirement is a feature of the system or a description of something the system is capable of doing in order to fulfill the system's purpose. First, we work with our customers to elicit the requirements, by asking

questions, demonstrating similar systems, and developing prototypes of all or part of the purposed system. Next, we capture those requirements in a document or database. The requirements are written first so that we and our customers agree on what the system should do. Then, the requirements are often rewritten, usually in a more mathematical representation, so that the designers can transform the requirements into a good design. A verification step ensures that the requirements are correct, and consistent, and a validation step makes sure that we have described what the customer intends to see in the final product.

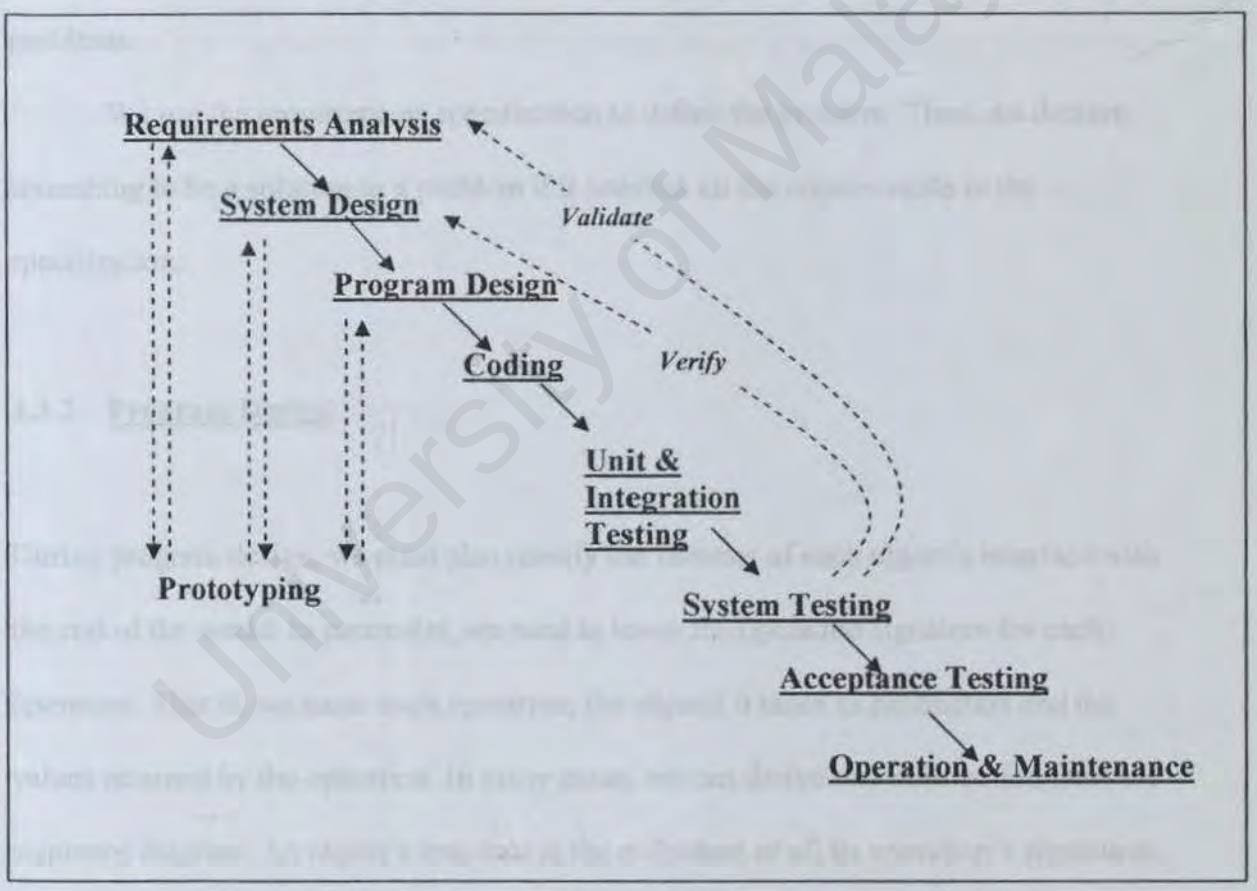


Figure 3.3.1 The waterfall model with prototyping.

3.3.2 System Design

Our customers usually want a new system either because there is no existing system or because there are undesirable aspects of the old system. In either case, the requirements documents tell us all about the problem that the system is to solve. Design is the creative process of transforming the problem into a solution; the description of a solution is also called design. As for this "Interactive 3D Human Head Multimedia System", the problems that rise can be quite a burden considering a whole wide of different people perspectives. System design is the best way to find the solution for these problems.

We use the requirements specification to define the problem. Then, we declare something to be a solution to a problem if it satisfies all the requirements in the specification.

3.3.3 Program Design

During program design, we must also specify the features of each object's interface with the rest of the world. In particular, we need to know the operation signature for each operation. That is, we name each operation, the objects it takes as parameters and the values returned by the operation. In many cases, we can derive this information from the sequence diagram. An object's interface is the collection of all its operation's signatures. Once we have defined the interfaces, we can classify them by type and build a hierarchy of interface types where some interfaces inherit properties from other interfaces. This

hierarchy has special importance, since objects are visible and accessible to other objects only through their interfaces.

3.3.4 Coding

Coding involves algorithms and data structures and the components are programming language primitives such as numbers, characters, pointers and control threads. In turn, there are primitive operators, including the language's arithmetic and data manipulation primitives, and composition mechanisms such as arrays, files and procedures. For this project, I use VRML (Virtual Reality Modeling Language) coding. You have to type the codes in text form, such as in the "Notepad" in Windows. Then you can view the models through a browser, the Internet Explorer plus a software called Cosmoplayer.

3.3.5 Unit & Integration Testing

In developing a system, testing usually involves several stages. First, each program component is tested on its own, isolated the other components in the system. Such testing, known as module testing, component testing or unit testing, verifies that the component functions properly with the types of input expected from studying the component's design. Unit testing is done in a controlled environment whenever possible, so the test team can feed a predetermined set of data to the component being tested and

observe what output actions and data are produced. In addition, the test team checks the internal data structures, logic, and boundary conditions for the input and output data. When collections of components have been unit-tested, the next step is ensuring that the interfaces among the components are defined and handled properly. Integration testing is the process of verifying that the system components work together as described in the system and program specification.

3.3.6 System Testing

Testing the system is very different from unit and integration testing. When you unit test your components, you have complete control over the testing process. You create your own test data, design your own test cases and run the tests yourself. When you integrate components, you sometimes work by yourself but often you collaborate with a small part of the test or development team. However, when you test a system, you work with the entire development team, coordinating what you do and being directed by the test team leader. The objective is to ensure that the system does what the customer wants it to do. To understand how to meet this objective, we first must understand where faults in the system come from.

3.3.7 Acceptance Testing

So far, all the tests have been run by the developers, based on their understanding of the system and its objectives. The customer also test the system, making sure that it meets

their understanding of the requirements, which may be different from the developers'. This test, called an acceptance test, assures the customers that the system they requested is the system that was built for them. The acceptance test is sometimes run in its actual environment but often is run at a test facility different from the target location. For this reason, we may run a final installation test to allow users to exercise system functions and document additional problems that result from being at the actual site.

3.3.8 Operation & Maintenance

When we develop systems, our main focus is on producing code that implements the requirements and works correctly. At each stage of development, I continually refer to work produced earlier stages. The design components ate tied to the requirements specification, the code components are cross-referenced and reviewed for compliance with the design, and the tests are based on finding out whether functions and constraints are working according to the requirements and design. Thus, development involves looking back in a careful, controlled way.

Maintenance is different. I look back at development products, but also at the present by establishing a working relationship with users and operators to find out how satisfied they are with the way the system works. I'm also looking forward, too, to anticipate things that might go wrong, to consider functional changes required by a changing business need, and to consider system changes required by changing hardware, software or interfaces.

3.3.9 Uses of Prototyping

1. Verifying user needs.
2. Verifying that design = specifications.
3. Selecting the "best" design.
4. Developing a conceptual understanding of novel situations.
5. Testing a design under varying environments.
6. Demonstrating a new product to upper management.
7. Implementing a new system in the user environment.

- **Observed Effects of Prototyping**

Software Product	Software Process
1. ease of use (+)	1. effort decreased (+)
2. user needs (+)	2. difficult cost-estimation (-)
3. unrealistic user expectations (-)	3. end-user participation increased (+)
4. added features (?)	more expertise needed (-)
5. poorer performance (-)	4. difficult planning & control (-)
6. mixed design quality	
7. mixed maintainability	
8. less need	
9. more difficult to do	

3.4 System Development Lifecycle

1. Systems Concept
2. System Design
3. System Development
4. System Implementation
5. System Support

This is just a summary of how I use to refer for my project-making. It'll help me a lot to understand the system life cycle for my project of an interactive 3D human head multimedia system. It consist a lot of information from the client or should I say the topic of my project and their ideas plus mine to form a complete system that is error-free. Below are figures simplified for better and easier understanding.

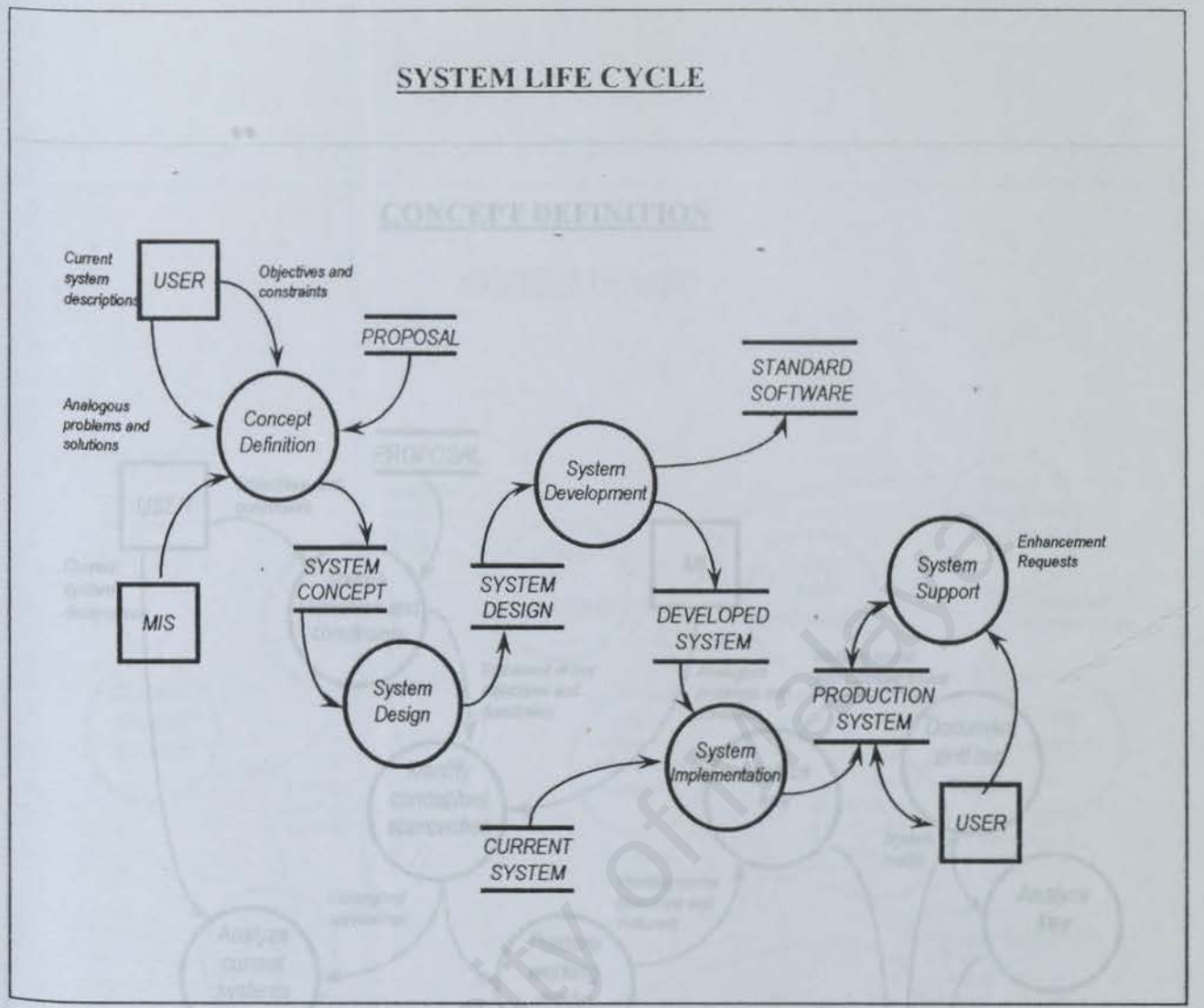


Figure 3.4.1 System life cycle.

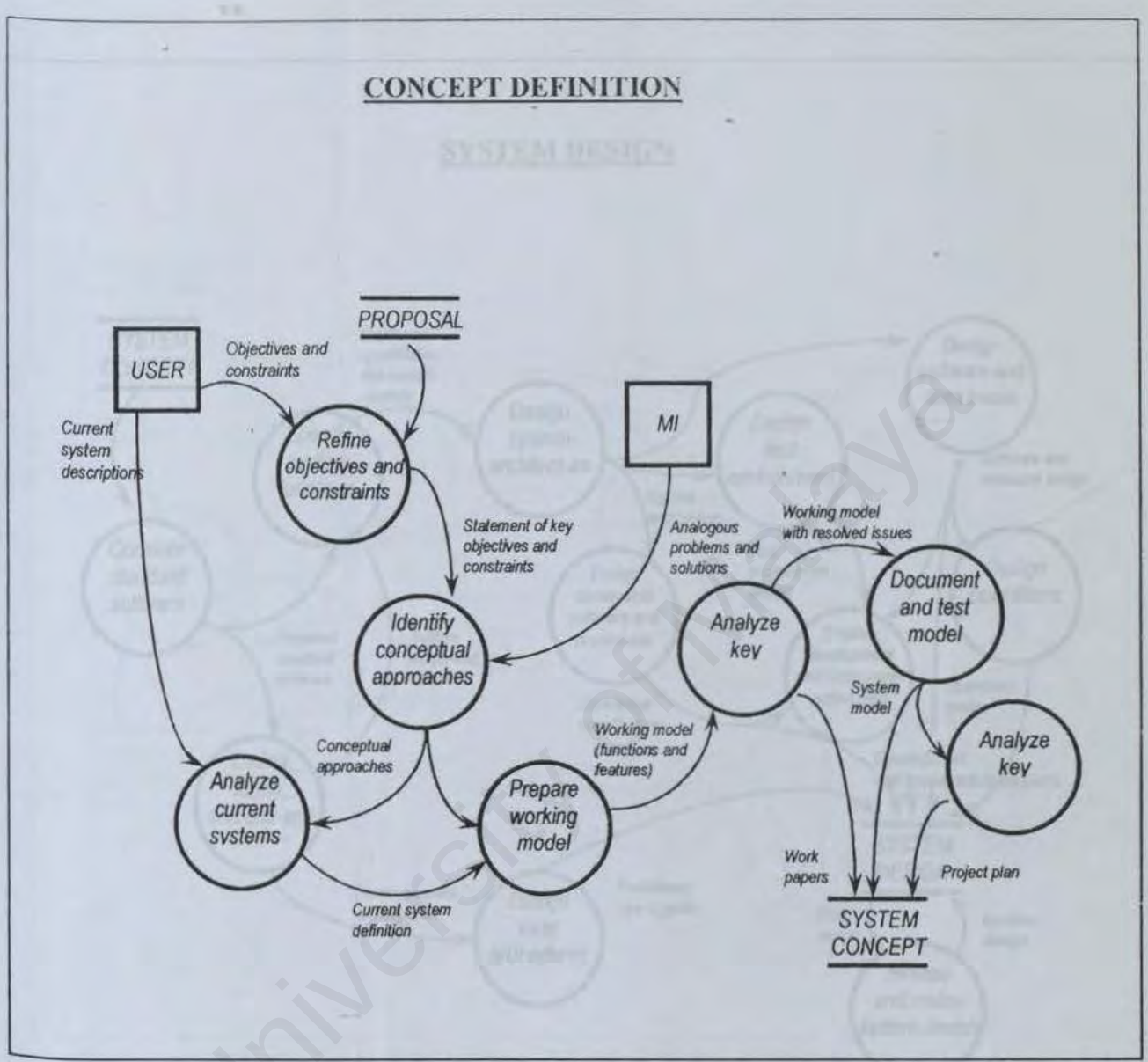


Figure 3.4.2 Concept definition.

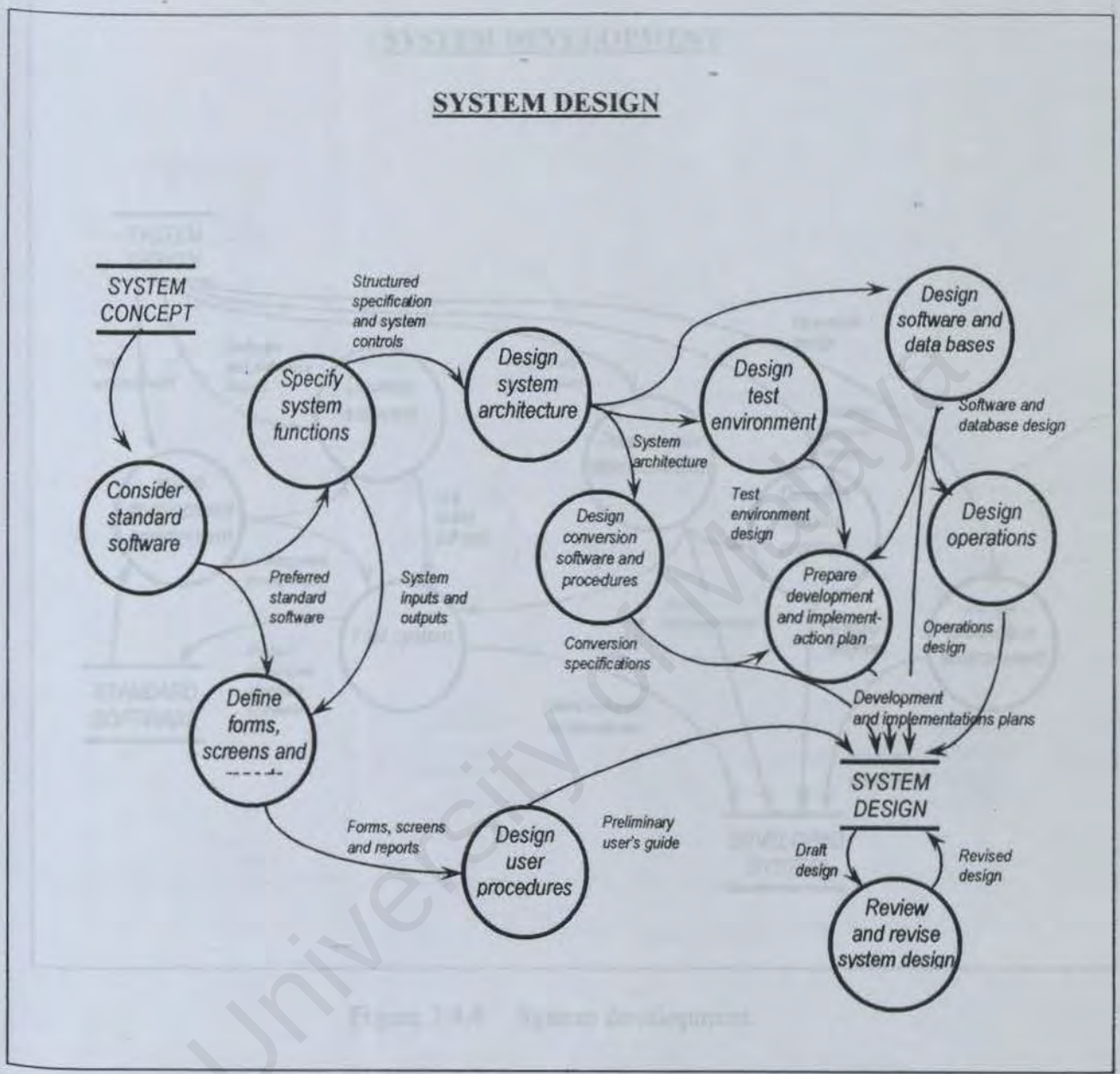


Figure 3.4.3 System design.

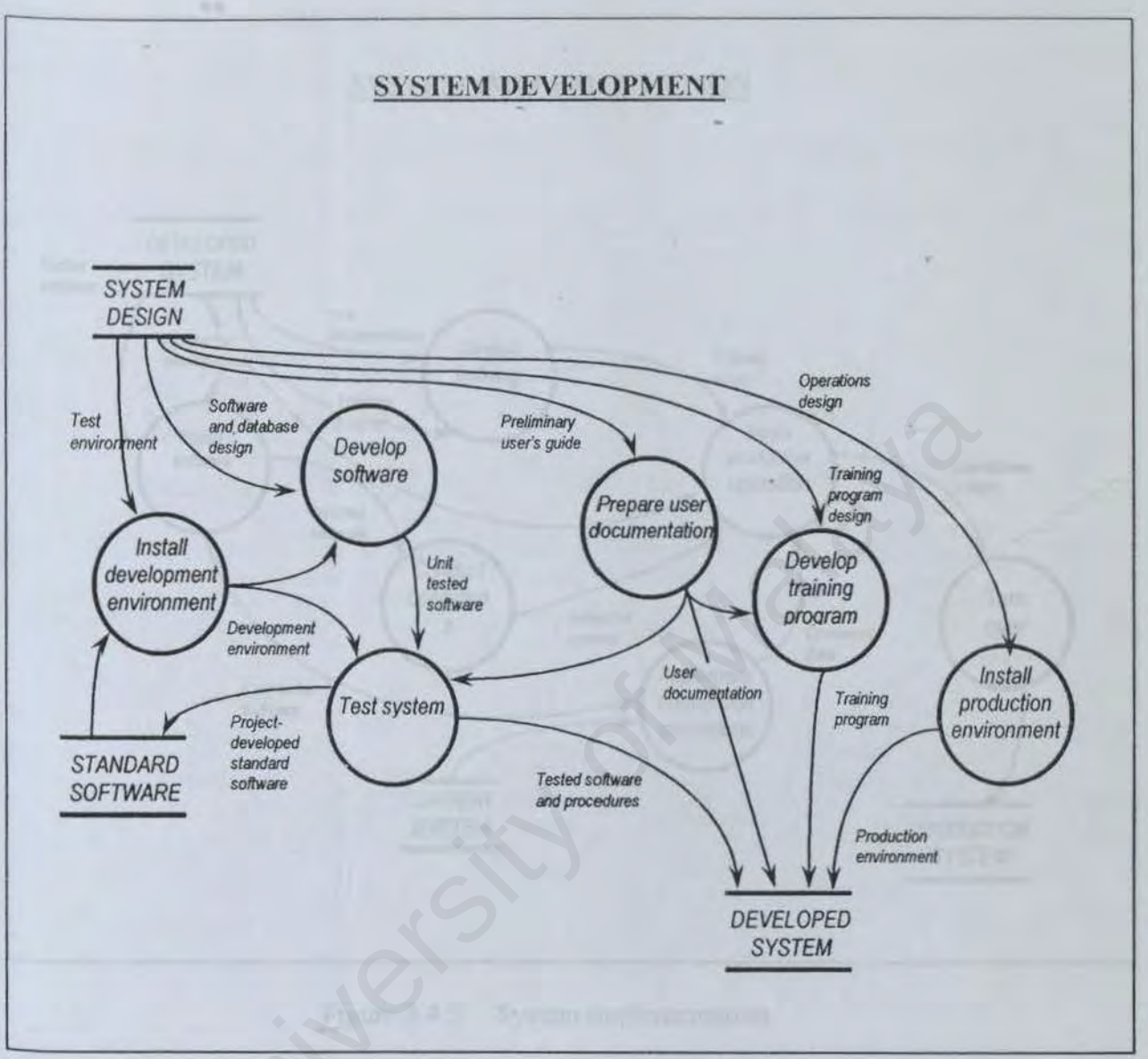


Figure 3.4.4 System development.

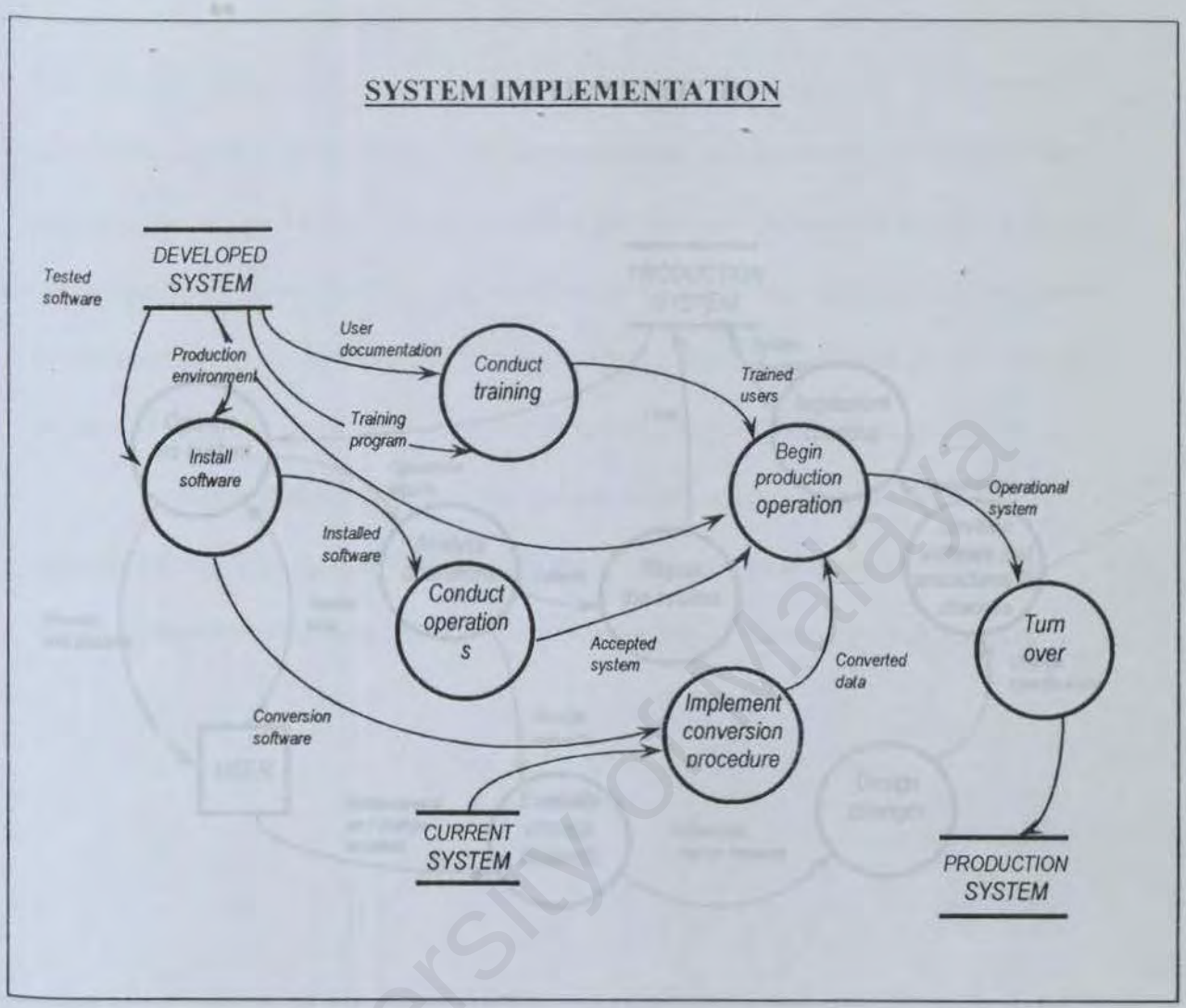


Figure 3.4.5 System implementation.

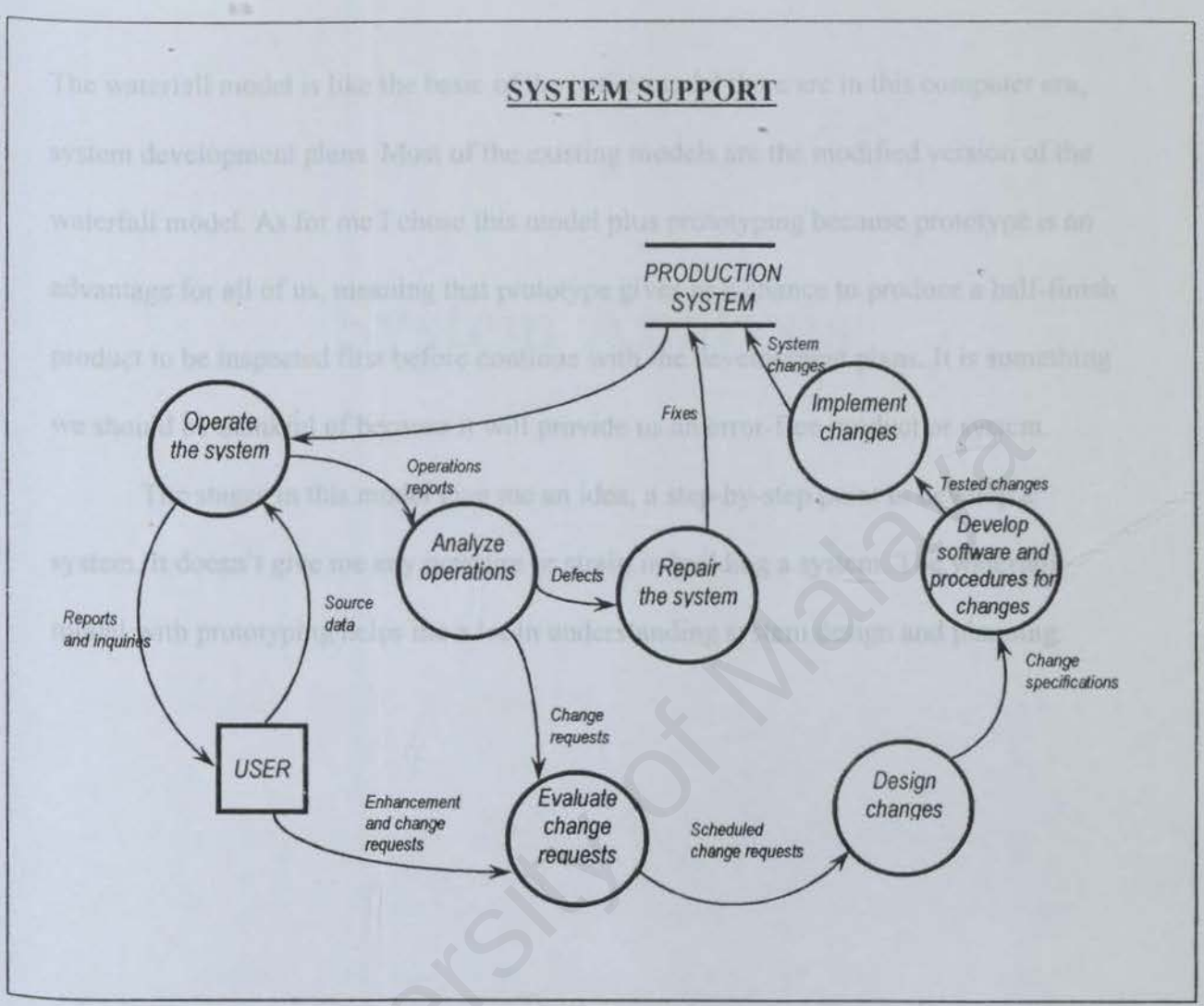


Figure 3.4.6 System support.

3.4 The Chosen Methodology

The waterfall model is like the basic of the entire model there are in this computer era, system development plans. Most of the existing models are the modified version of the waterfall model. As for me I chose this model plus prototyping because prototype is an advantage for all of us, meaning that prototype gives us a chance to produce a half-finish product to be inspected first before continue with the development plans. It is something we should be thankful of because it will provide us an error-free product or system.

The stages in this model give me an idea, a step-by-step point to develop a system. It doesn't give me any pressure or strain in building a system. The waterfall model with prototyping helps me a lot in understanding system design and planning.

System Analysis

4.1 System Analysis

For this chapter of my project, I use my knowledge of computers and computing to help solve problems. Often the problem with which I am dealing is related to a computer or an existing computer system, in this case I am building a system of an interactive 3D human head multimedia system, but sometimes the difficulties underlying the problem have nothing to do with computers. Therefore, it is essential that I first understand the nature of the problem. In particular, I must be very careful not to impose computing machinery or techniques on every problem that comes my way. I must solve the problem first. Then, if need be, I can use technology as a tool to implement my solution. My analysis has shown that some kind of computer system is necessary or desirable to solve a particular problem at hand.

Most problems are large and sometimes tricky to handle, especially if they represent something new that has never been solved before. So we must begin investigating it by analyzing it, that is, by breaking the problem into pieces that we can understand and try to deal with. We can thus describe the larger problem as a collection of small problems and their interrelationships. It is important to remember that the relationships are as essential as the sub problems them selves. Sometimes, it is the relationships that hold the clue to how to solve the larger problem, rather than simply the nature of the sub problems.

Once I have analyzed the problem, I must construct my solution from components that address the problem's various aspects. Synthesis, the reverse process, is the putting together of a large structure from small building blocks. As with analysis, the

composition of the individual solutions may be as challenging as finding the solutions themselves. To see why, consider the process of writing a novel. The dictionary contains all the words that you might want to use in your writing. But the most difficult part of writing is deciding how to organize and compose the words into sentences, and likewise the sentences into paragraphs and chapters to form the complete book. Thus, any problem-solving technique must have two parts: analyzing the problem to determine its nature, and then synthesizing a solution based on our analysis.

4.2 Functional Requirements

Requirements describe a system's behavior. As the system acts on data or instructions, objects or entities move from one state of being to another: from empty to full, from busy to still, or from sending to receiving, for example. That is, in any given state, the system satisfies a set of conditions; when the system acts, it may change its overall state by changing the state of an object. The requirements express the system and object states and the transitions from one state to another. In particular, the requirements describe the activities of the system, such as a reaction to input, and the state of each entity in the system before and after the activity occur.

To help describe requirements, we can think of them in two ways: functional and nonfunctional. A functional requirement describes an interaction between the system and its environment. For example, to determine functional requirements, I decide what states are acceptable ones for the system to be in. Further, functional requirements describe how the system should behave given certain stimuli.

- Description of an interaction between the system and its environment.
- Topics or issues relevant to what the users want to find in the system.
- What kind of input the users want to put in and how the output will be resulted.
- The type of output or answer, the users looking for.
- An easy and user-friendly system and interfaces.
- Concepts understanding between the users and the system.
- Screen to display description texts.
- An option to move to function page and skip description page.
- Let users chose the topics in which category they want; freedom in surfing.

4.3 Nonfunctional Requirements.

The questions addressed by functional requirements have answers that are independent of an implementation of a solution to the customer's problem. We (the clients or the topic of this project and me) describe what the system will do without discussing the particular computer we might use, the programming language employed, the internal data structures involved, or the kind of end product will be stored. Rather than telling us what the system will do, these requirements put restrictions on the system. That is, a nonfunctional requirement or constraint describes a restriction on the system that limits our choices for constructing a solution to the problem.

Both functional and nonfunctional requirements are elicited from the customer in a formal, careful way, in this case I have to do some research and studies on the related topic that is a human head anatomy, considering the target users and its purposes. This

formal requirements elicitation is necessary because the target users or the customers are not always good at describing exactly what they want or need, and I am not always understanding someone else's business concerns. Thus, if I'm not carefully organized and encouraged, the communication between the customers can lead to misunderstanding or incomplete specification. I could misinterpret the objectives and the scopes of this project.

- There are still limitations for virtual reality to include smell, touching but as for surgery simulation, it is not impossible and yet it is not 100% recommended when it is perform onto an actual human being.
- The type of language that I use for the system. Important for globalization. (Icons, interfaces, colors etc)
- The type of software that the users going to use when downloading files or images. (for example that the users must have Cosmoplayer to browse the models – they can download it from the internet)
- The implementation techniques and tools that I use to design and setup this website.
- Medical sciences on human head anatomy are vast and keep expanding. (up-to-date information will be delay – limited documentation)
- Limitation on human head anatomy descriptions and functionalities.

4.4 Software Requirements

For this project I specifically chose quite a few choices of software that I will use in my system development soon. This is important to understand each functions, the purpose of the software that being use. It is vital for the success of my project. Let's see what I have in store, I mean which software I will be using in the text below.

4.4.1 Virtual Reality Modeling Language (VRML 2.0)

The Virtual Reality Modeling Language (VRML) is a file format for describing 3D interactive worlds and objects. It may be used in conjunction with the World Wide Web. It may be used to create three-dimensional representations of complex scenes such as illustrations, product definition and virtual reality presentations.

VRML has been designed to fulfill the following requirements:

Authorability

Make it possible to develop application generators and editors, as well as to import data from other industrial formats.

Completeness

Provide all information necessary for implementation and address a complete feature set for wide industry acceptance.

Composability

The ability to use elements of VRML in combination and thus allow re-usability.

Extensibility

The ability to add new elements.

Implementability

Capable of implementation on a wide range of systems.

Multi-user potential

Should not preclude the implementation of multi-user environments.

Orthogonally

The elements of VRML should be independent of each other, or any dependencies should be structured and well defined.

Performance

The elements should be designed with the emphasis on interactive performance on a variety of computing platforms.

Scalability

The elements of VRML should be designed for infinitely large compositions.

Standard practice

Only those elements that reflect existing practice, that are necessary to support existing practice, or that are necessary to support proposed standards should be standardized.

Well-structured

An element should have a well-defined interface and a simply stated unconditional purpose. Multipurpose elements and side effects should be avoided.

VRML is capable of representing static and animated objects and it can have hyperlinks to other media such as sound, movies, and image. Interpreters (browsers) for VRML are widely available for many different platforms as well as authoring tools for the creation VRML files.

VRML supports an extensibility model that allows new objects to be defined and a registration process to allow application communities to develop interoperable extensions to the base standard. There is a mapping between VRML elements and commonly used 3D application programmer interface (API) features.

The scope of the standard incorporates the following:

- a mechanism for storing and transporting two-dimensional and three-dimensional data
- elements for representing two-dimensional and three-dimensional primitive information
- elements for defining characteristics of such primitives
- elements for viewing and modeling two-dimensional and three-dimensional information
- a container mechanism for incorporating data from other metafile formats

- mechanisms for defining new elements which extend the capabilities of the metafile to support additional types and forms of information

SETTING UP THE ENVIRONMENT

I have used the following set up: Windows 98 and Windows XP with Communicator 4.04 and Cosmo player 2. The set up has to be precise so the principals and steps taken here will probably transfer to other platforms but no guarantees are given. You need to download both of the above packages and install them. This is done with the automatic winstall and you should let the install set up the paths.

The .wrl file, or VRML file.

The VRML file is slightly more complicated, although not at first glance:

```
#VRML V2.0 utf8

DEF Root Transform {}
```

To understand this you need to know something about VRML and the naming conventions. VRML allows the user to define nodes using the DEF statement before the node and a string that will then be associated with that node. Within a VRML scene the node can then be reused by using the USE statement. So in the above world there is one node, a transform, which is named root. This does appear to give you very little versatility, which is you appear to be only able to access and affect those nodes that are pre-defined. However those nodes added by the applet are also accessible through there object names rather than there their DEF names.

Appearance node

A node of type Appearance, FontStyle, ImageTexture, Material, MovieTexture, PixelTexture, or TextureTransform. Appearance nodes control the rendered appearance of the geometry nodes with which they are associated.

Bindable leaf node

A node of type Background, Fog, NavigationInfo, or Viewpoint. These nodes may have many instances in a scene graph, but only one instance may be active at any instant of time.

Children nodes

Nodes which are parented by grouping nodes and thus are affected by the transformations of all ancestors. See "Concepts - Grouping and Children Nodes" for list of allowable children nodes.

Color model

Characterization of a color space in terms of explicit parameters. VRML allows colors to be defined only with the RGB color model.

Display device

A graphics device on which VRML scenes can be represented.

Drag sensor

Drag sensors (CylinderSensor, PlaneSensor, SphereSensor) cause events to be generated in response to pointer motions which are sensor-dependent. For example, the SphereSensor generates spherical rotation events. See "Concepts - Drag Sensors" for details.

Event

Messages sent from one node to another as defined by a Route. Events signal changes to field values, external stimuli, interactions between nodes, etc.

Exposed field

A field which can receive events to change its value(s) and generates events when its value(s) change.

Execution model

The characterization of the way in which scripts execute within the context of VRML.

External prototype

Prototypes defined in external files and referenced by a URL.

Field

The parameters that distinguish one node from another of the same type. Fields can contain various kind of data and one or many values.

Geometry node

Nodes of type Box, Cone, Cylinder, ElevationGrid, Extrusion, IndexedFaceSet, IndexedLineSet, PointSet, Sphere, and Text which contain mathematical descriptions of three-dimensional points, lines, surfaces, text strings and solid objects.

Geometric property node

A node of type Color, Coordinate, Normal, or TextureCoordinate. These nodes define the properties of specific geometry nodes.

Geometric sensor node

A node of type ProximitySensor, VisibilitySensor, TouchSensor, CylinderSensor, PlaneSensor, or SphereSensor. These nodes generate events based on user actions, such as a mouse click or navigating close to a particular object.

Grouping node

A node of type Anchor, Billboard, Collision, Group, or Transform. These nodes group child nodes and other grouping nodes together and cause the group to exhibit special behavior which is dependent on the node type.

IETF

Internet Engineering Task Force. The organization which develops Internet standards.

Instance

An instantiation of a previously defined node created by the USE syntax.

Interpolator node

A node of type ColorInterpolator, CoordinateInterpolator, NormalInterpolator, OrientationInterpolator, PositionInterpolator, or ScalarInterpolator.These nodes define a piece-wise linear interpolation of a particular type of value at specified times.

JPEG

Joint Photographic Experts Group.

MIDI

Musical Instrument Digital Interface - a standard for digital music representation.

MIME

Multipurpose Internet Mail Extension used to specify file typing rules for browsers. See "Concepts - File Extension and MIME Types" for details.

Node

The fundamental component of a scene graph in VRML. Nodes are abstractions of various real-world objects and concepts. Examples include spheres, lights, and material descriptions. Nodes contain fields, and events. Messages are sent between nodes via routes.

Node type

A required parameter for each node that describes, in general, its particular semantics. For example, Box, Group, Sound, and SpotLight. See "Concepts - Nodes, Fields, and Events" and "Nodes Reference" for details.

Prototype

The definition of a new node type in terms of the nodes defined in this standard.

RGB

The VRML color model. Each color is represented as a combination of the three primary colors red, green, and blue.

Route

The connection between a node generating an event and a node receiving an event.

Scene graph

An ordered collection of grouping nodes and leaf nodes. Grouping nodes, such as Transform, LOD, and Switch nodes, can have child nodes. These children can be other grouping nodes or leaf nodes, such as shapes, browser information nodes, lights, viewpoints, and sounds.

Sensor node

A node of type Anchor, CylinderSensor, PlaneSensor, ProximitySensor, SphereSensor, TimeSensor, TouchSensor, or VisibilitySensor. These nodes detect changes and generate events. Geometric sensor nodes generate events based on user actions, such as a mouse click or navigating close to a particular object. TimeSensor nodes generate events at regular intervals in time.

Special group node

A node of type LOD (level of detail), InLine, or Switch. These nodes are grouping nodes which exhibit special behavior, such as selecting one of many children to be rendered based on a dynamically changing parameter value or dynamically loading its children from an external file.

Texture coordinates

The set of 2D coordinates used by vertex-based geometry nodes (e.g. IndexedFaceSet and ElevationGrid) and specified in the TextureCoordinate node to map textures to the vertices of some geometry nodes. Texture coordinates range from 0 to 1 across the texture image.

Texture transform

A node which defines a 2D transformation that is applied to texture coordinates.

URL

Uniform Resource Locator as defined in IETF RFC 1738.

URN

Uniform Resource Name

VRML document server

An application that locates and transmits VRML files and supporting files to VRML client applications (browsers).

VRML file

A file containing information encoded according to this standard.

4.4.2 Cosmoplayer

Open the door to Virtual Reality with Cosmo Player. Cosmo Player is a high-performance, cross-platform VRML 2.0 client designed for fast and efficient viewing of virtual worlds. Navigate and manipulate 3D scenes and bring your Web experience to a new level.

Use VRML to fly through anatomy class, experience 3D data visualizations, or show off a CAD model. Cosmo Player is the premiere viewing client for VRML, with support for the latest standards.

Whether on the Internet or in an enterprise, Cosmo Player allows web content creators and applications developers to add visual and multimedia elements to their work.

4.4.3 3D Studio Max 5

3ds max 5 - Discreet's latest release of the world's most widely-used professional 3D modeling, animation and rendering solution - contains all the essential tools required for creating eye-catching animation, cutting-edge games, and distinctive design visualizations.

New feature highlights

- Multiple global Illumination solvers
- Vertex weight table
- Significant polygon modeling enhancements
- Unwrap UVW
- Render to texture
- Set Key animation system
- Revamped curve editor
- Photometric lights
- Dope sheet editor
- Ink 'n paint
- Weighted controllers
- Draw curves
- Interactive dynamics
- Auto tangents

Animation

- Unique, weighted Animation Controller Subsystem allows for multiple algorithms to drive any animation channel (including custom expressions)
- Controllers can be layered, blended, scripted, referenced and instanced

- Advanced animation controllers include reactors for event-driven animation, blocks for reusing animation clips in non-linear fashion, and expressions for establishing dynamic relationships between parameters
- The Midi motion capture controller enables ability to drive animation with a midi-enabled input device
- Flex modifier enables soft-body dynamics with option for collision detection
- Curve editor and dope sheet control every animated parameter with filtering to isolate relevant data
- Audio Waveforms displayed in track bar and curve editor
- Key management includes Bezier function curves, ease and multiplier curves, out-of-range animation control, key frame reduction, waveform support, constant velocity, text notes, time tags, move/scale/randomize time segments and sub-frame editing
- Character tools include volumetric skinning with hierarchies or splines, spring-based secondary animation, progressive morphing, and FFD lattices
- Linked X-form Modifier allows sub-object components to be uniformly controlled by null objects
- IK results can be calculated or interactive
- Both History-Independent and History-Dependent Limb and Spline IK solvers
- Integrated particle systems with behavioral control including inter-particle collision, meta-particles, snow, spray, bubbles, explosions, spawning, and trails
- Bones support Squash and Stretch
- Rigid body dynamics for colliding and sliding objects
- Schematic view of complex hierarchies

- Space warps provide anima table, world-space forces and deformation algorithms
- Enhanced angle deformers including joint, morph, and bulge for precise creation of animation extents based on joint rotation
- Parameter wiring easily connects any anima table parameter of one object to another
- Display animation proxies to simplify screen-based geometry for animation and dynamics simulations – yet retain full high-quality object definition at render time
- Dynamics core with estimated momentum and inertia properties for keyframed objects, allowing full interaction of keyframed and dynamically-simulated objects
- Cloth fully interacts with fluid dynamics, giving full control over the cloth air resistance and relative density for floating objects
- Full collision detection with Hard and Soft Body dynamics
- Constrain cloth, soft bodies, or rope to other dynamics-driven or keyframed objects
- Construct automated machinery with the dynamics motor that allows arbitrary rotation and gain to be applied to any object
- Objects may be assembled to act as a single, rigid body simulation, but instructed to break apart if they hit another object with sufficient force
- Interactive Dynamics Preview supports Open GL and Direct 3D, and includes many scene properties with lighting and texturing

Modeling

- Extensive set of 2D and 3D primitives can remain parametric or convert to any other base geometry

- Comprehensive tools for modeling splines, polygons, polygonal mesh, Bezier patch or relational NURBS Surfaces in direct or procedural mode
- Advanced spline-based patch modeling with exhaustive polygonstyle editing tools
- Support vertex colors, illumination and alpha with patches for real-time games applications
- Rapid mesh modeling in either explicit or procedural modes with extensive set of vertex, edge, face and polygon tools that include cut, chamfer, bevel, divide, slice, planarity control, interactive normal flipping, and local tessellation
- Renderable splines enable more efficient creation of ropes, cables, etc.
- Hierarchical Sub-division surfaces provide state-of-the-art localized subdivision
- Relational NURBS modeling creates curves and surfaces which maintain design intent through manipulation and animation with choice of point or CV curves and surfaces
- NURBS surfaces include point and CV, u-loft, uv-loft, blend, n-sided blend, offset, extrude, fillet, lathe, ruled, cap, 1 rail, 2 rail, trim, and multi-curve trim
- NURBS curves include point and CV, fit, offset, chamfer, fillet, and surface-surface intersection, surface edge, iso, curve on surface, and projected with trimming control
- Mesh smooth provides intuitive subdivision surface modeling including modifiable points, vertex and edge weighting, and interactive control of mesh tessellation
- Soft Selections on polygons, patches, and splines

Rendering

- High speed film-quality renderer provides 16-bit color per channel scan line A-buffer, with full gamma control at up to 32K lines of resolution for frames or fields
- Superior scalability through multi-threading and free customizable network rendering
- Selective ray tracing provides fast, accurate reflections and refractions with very high recursion levels
- Advanced rendering options include photo-realistic depth of field, adaptive displacement of all geometries, 2D/3D motion blur, 3D volumetric lighting, fire, explosions, smoke and fog
- Render Effects delivers real-time photo-realistic results for effects including blur, depth of field, glow, film grain, lens flare highlights and color correction
- Render objects as in reflections, only without being present in the scene
- Over a dozen anti-aliasing filters, providing different looks for rendered images including Area, Blackman, Catmull-Rom, Soften, and more
- Direct plug-in control of key rendering stages including anti-aliasing, shaders, sampling, and shadows
- Render Elements provides multiple render outputs (specular, diffuse, reflection, shadow, and more) from a single rendering pass
- Exposure controls simulate the way the human eye reacts to extreme lighting conditions within the same scene
- Full Radiosity simulation with photometric data support
- ActiveShade provides efficient workflow for tweaking scene lighting and materials without full re-renders

Lights

- Light types: omni, free and target spot, free and target directional, and photometric lights
- Controls include: shadows, shadow color and density, projected images, contrast, edge softness, attenuation and decay
- Isolate ambient/diffuse/specular control
- Full Volumetric Lighting supported
- Interactive glows, flares, streaks and highlights

Cameras

- Unlimited number of cameras using industry-standard camera types with optional custom relationships
- Interactive clipping plane, dolly, FOV, grid overlay, orbit, roll, vertigo zoom, zoom and safe frame display
- Precise alignment with either horizontal, vertical, or diagonal field-of-view measure, and support for orthogonal projection

Materials and mapping

- Combine an unlimited number of textures to give ultimate control over materials

- Material Map Browser portrays hierarchies with thumbnails and drag-and-drop assignment
- Shaders provided include Anisotropic, Blinn, Oren-Nayar-Blinn, Phong, Metal, Multi-layer, and Strauss with independent sampler options
- Over 30 procedural 2D and 3D maps provided
- Multiple UVW mapping with up to 100 mapping channels per vertex
- Vertex colors can be painted, tinted or derived from scene lighting and shadows
- Mapping projections include procedural, planar, cylindrical, spherical, box, face, shrink-wrap, world-XYZ, camera, and screen
- Direct manipulation of texture vertices with the extensive UVW
- Unwrap modifier
- Translucency material simulates effect of light passing through an object
- Shellac material gives artists ability to combine shading algorithms for advanced material definition
- Matte/Shadow Material enables scenes to be prepared for compositing
- Extensibility
- Plug-in architecture provides extensibility for nearly any system component
- Plug-ins behave like core features to support any new functionality introduced
- Bundled Software Developer's Kit (SDK) enables c++ developers to extend the software in countless ways
- Customizable interfaces include interface elements, colors, custom toolbars, buttons, tool tips, macros and scripts
- Viewport interaction

- WYSIWYG viewport environment with multi-textures per face, blended true transparency, Phong highlights, procedurals, and display of Pixel and Vertex shaders
- Support for Dual Planes buffering
- Multiple coordinate systems: View, Screen, World, local, chosen object, grid, or parent space
- Interactive axis constraints and modeless keyboard entry supported 3D snap system with more than 20 snap types
- Align system for positioning selections and highlights
- Selection methods include pick, fence, rectangular, lasso, and circle with object class filters, select-by-name, named selection sets, mesh extents, back face exclusion, polygon boundary, smoothing, material ID, and face normal angle
- Schematic view for controlling scene, object, modeling, and material relationships
- In-viewport Motion Blur and Depth Of Field
- User-definable scene scale

Architecture

- Multi-threaded throughout for superior performance and scalability
- Scripting Language implemented at the core level
- Supports OpenGL and Direct3D hardware acceleration, and Heidi® software acceleration for any Windows® display
- Flexible procedural modeling stores decisions as long as required
- Plug-in classes

- Object types: 3D and 2D base geometry classes, arithmetic objects, particle systems, animation systems, space warps, helpers
- Modeling operations: modifiers may be parametric or explicit and may behave in either object or world space
- Animation functions: controllers (for parameters, matrices, or systems), motion capture devices, utilities, sound, key/time manipulation
- Image effects: layer, compositing, transition, one pass, image I/O, and interactive rendering effects
- Scene interaction: object snaps, color pickers, utilities, user interfaces, DCOM application control
- Rendering: complete renderers, anti-aliasing, shaders, samplers, environments, shadows, lights, cameras, materials, 2D or 3D procedural, composite, or explicit textures
- File I/O: geometry, scene, bitmap, image device, fonts, viewer
- Scripting
- MAXScript object-oriented scripting language mirrors SDK to provide access to plug-in parameters
- Scripting generates seamless interfaces you can load, launch at startup, or embed in files
- Dynamic macro recording creates concise scripts in MAXScript syntax in relative or explicit mode
- Plug-in scripts can append to plug-ins, abstract plug-ins into alternative interfaces, or combine several plug-ins in one interface

- Visual scripting language interface for MAXScript: add user interface elements like sliders, dropdown lists, check boxes, buttons, color pickers, etc., to any object, modifier or material
- Link to parameters using parameter wiring
- Workflow
- Scenes are self-contained definitions of objects' animation and rendering choices
- External references allow single objects/characters or entire scenes to be referenced
- Undo- and Redo-definable in-depth with separate scene and view port lists
- Context-sensitive Quad Menus deliver fast and efficient workflow
- Share data with Autodesk VIZ® for design visualization projects
- Superior 2D/3D integration with Discreet combustion®, inferno®, flame®, flint® effects solutions and fire® and smoke® non-linear editing system

System requirements

- Intel® or AMD® based processor at 300 Mhz minimum (Dual Intel Pentium® 4 processor or dual AMD Athlon™ system recommended)
- 256MB RAM and 300MB swap space minimum (1GB RAM and 2GB swap space recommended)
- Graphics card supporting 1024 x 768 x 16-bit color (OpenGL and Direct3D hardware acceleration supported; 24-bit color, 3D graphics accelerator preferred)
- Windows-compliant pointing device (specific optimization for Microsoft® Intellimouse™)

- DVD or CD-ROM drive
- Optional: sound card and speakers, cabling for TCP/IP-compliant network, 3D hardware graphics acceleration, video input and output devices, joystick, midi-instruments, 3-button mouse

4.4.4 Dreamweaver MX

For the first time, I can work in a single integrated environment to create, build, and manage websites and Internet applications. Macromedia Dreamweaver MX combines its renowned visual layout tools with the rapid web application development features of Dreamweaver UltraDev and the extensive code-editing support of Macromedia HomeSite. So the world's best way to create professional websites is now the easiest way to build powerful Internet applications.

System Requirement

- 96 MB of free available system RAM (128 MB recommended)
- 275 MB of available disk space

- 256 color monitor capable of 800 x 600 resolutions (1024 x 768, millions of colors recommended)

- Intel Pentium II or equivalent 300+ MHz processor
- Windows 98 SE, Windows Me, Windows NT 4, Windows 2000, or Windows XP

4.4.5 Adobe Photoshop 7.0

Adobe® Photoshop® 7.0 software helps you work more efficiently, explore new creative options, and produce the highest quality images for print, the Web, and anywhere else. Sophisticated painting tools including brushes that simulate natural media. Industry-standard pen tool for precision drawing. Pattern Maker to generate background textures automatically. Layer effects for instant — and editable — effects. Layer styles to apply multiple effects instantly. More than 95 special effects filter. Liquify tools and Turbulence brush to distort images with precision. Powerful color correction tools. Healing brush to remove flaws while preserving tonality and texture. Layers for editable compositing. Picture Package tools to print a folder of images quickly. Sophisticated crop tool that corrects perspective as it crops.

4.4.6 Macromedia Flash MX

Macromedia Flash MX is the "must have" tool used by over one million professionals to deliver the best user experiences on the web, increasing both revenue and customer satisfaction while lowering costs. The approachable environment includes powerful video, multimedia, and application development features, which allow designers and developers to create rich user interfaces, online advertising, e-learning courses and enterprise application front ends. Macromedia Flash MX enables the best Internet experiences that are faster and more familiar. Macromedia Flash MX is a powerful environment for creating a broad range of high-impact content and rich applications for the Internet. Deliver real results with improved return on investment through better user experiences, lower bandwidth costs, and increased developer productivity. Increase traffic, drive revenue and secure long-term customer relationships: Both designers and developers can now create rich content and applications that provide a competitive advantage on the web. Take advantage of Internet application development trends by using templates and pre-built components to rapidly prototype application interfaces. Download MP3 and JPEG files at runtime and save download and production time.

4.4.7 Sound Forge Studio 6.0

Sound Forge Studio includes much of the same award-winning technology found in Sound Forge, my choice of professional digital audio editor. This streamlined version makes it easy to record, edit, and process audio using your PC. Record from a CD, microphone, LP, cassette, or musical instrument, and then edit your recording using a wide range of tools and effects. Liven up presentations, recordings, and music using over 30 digital audio effects and processes, including 1001 Sound Effects. I can create my own "Best Of" audio CDs using track-at-once CD burning. Platform: Windows 98SE, Me, 2000, or XP. I can do a lot of interesting thing using SoundForge such as audio recording and editing, effects processing CD, ripping and burning and audio encoding

4.5 Hardware Requirement

Processor: Intel Pentium III 833 MHz

RAM: 128 Mb

Hard Disk Capacity: 20 Gb

CD-ROM: Acer CDRW 10 x 8 x 32

OS: Windows XP

Monitor: Acer 14"

Sound Card: VIA

Graphic Card: NVIDIA RIVA TNT2

This is what I'm currently using, the specification of my computer. It would do a lot of fascinating thing such as, modeling, rendering, compiling, editing and many more if the current specification is higher or greater.

System Design

5.1 Conceptual design / System Design

Design is the creative process of transforming the problem into a solution; the description of a solution is also called design. I use the requirements specification to define the problem. Then, I declare something to be a solution to a problem if it satisfies all the requirements in the specification. In many cases, the number of possible solutions is limitless. The nature of the solution may change as the solution is described or implemented. The description of a system may change during the development cycle. System design or conceptual design describes the system in language that the customer and I can understand, rather than in jargon and technical terms. The conceptual design may even list acceptable user responses and the actions that may result.

What do we have on the human head? The face, the nose, the mouth, the brain, the ears and the internal organs. I have to design the whole feature and visualize it. And then apply it onto the modeling tools. Each step has to be carefully recognized and applied. I have to understand the human head anatomy and its functions. By doing this, I have to do research on human head anatomy and carefully selected the best information for my target users. I'm going to explain this a little bit further towards the end of this chapter.

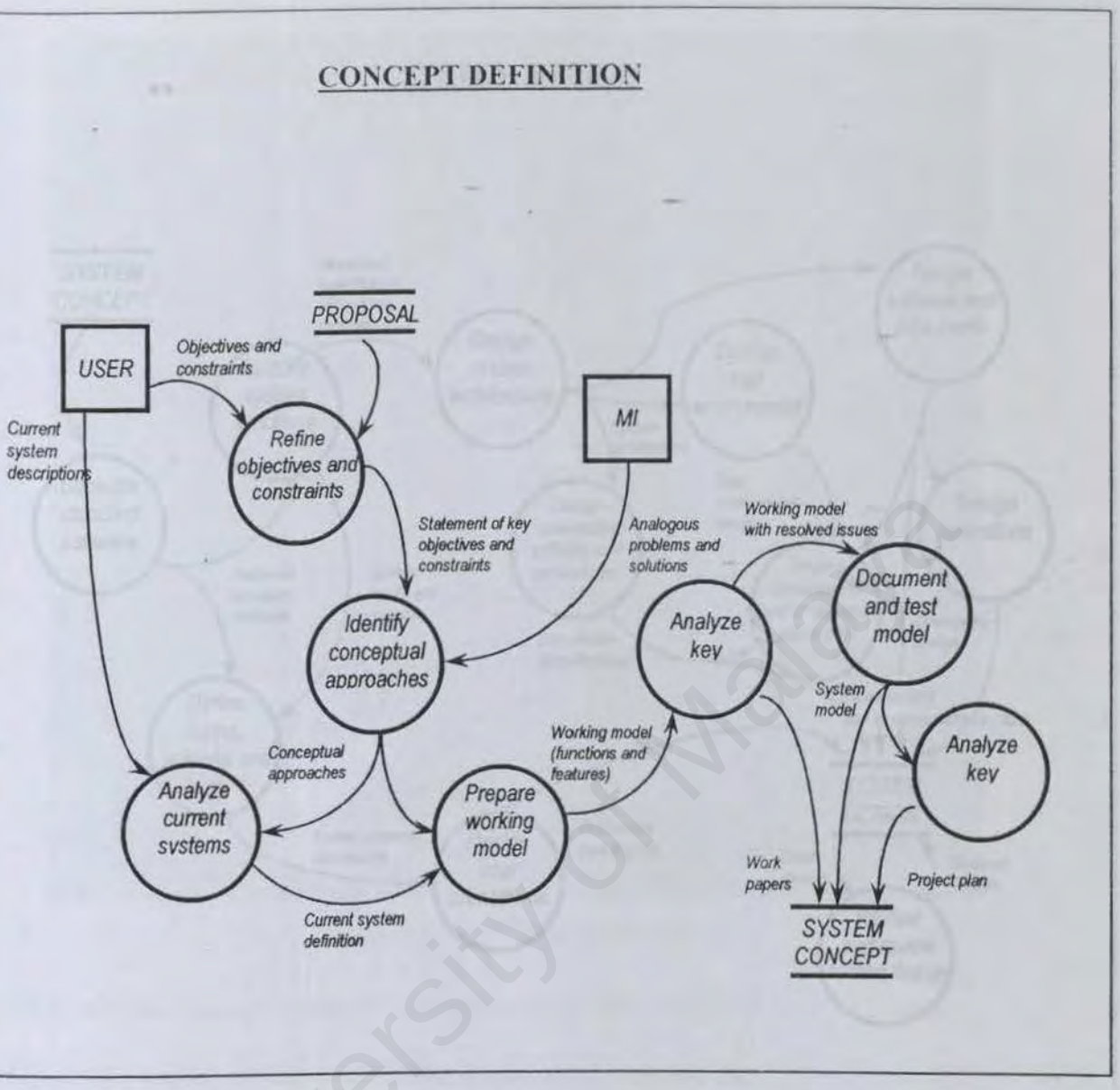


Figure 5.1.1 Concept definition.

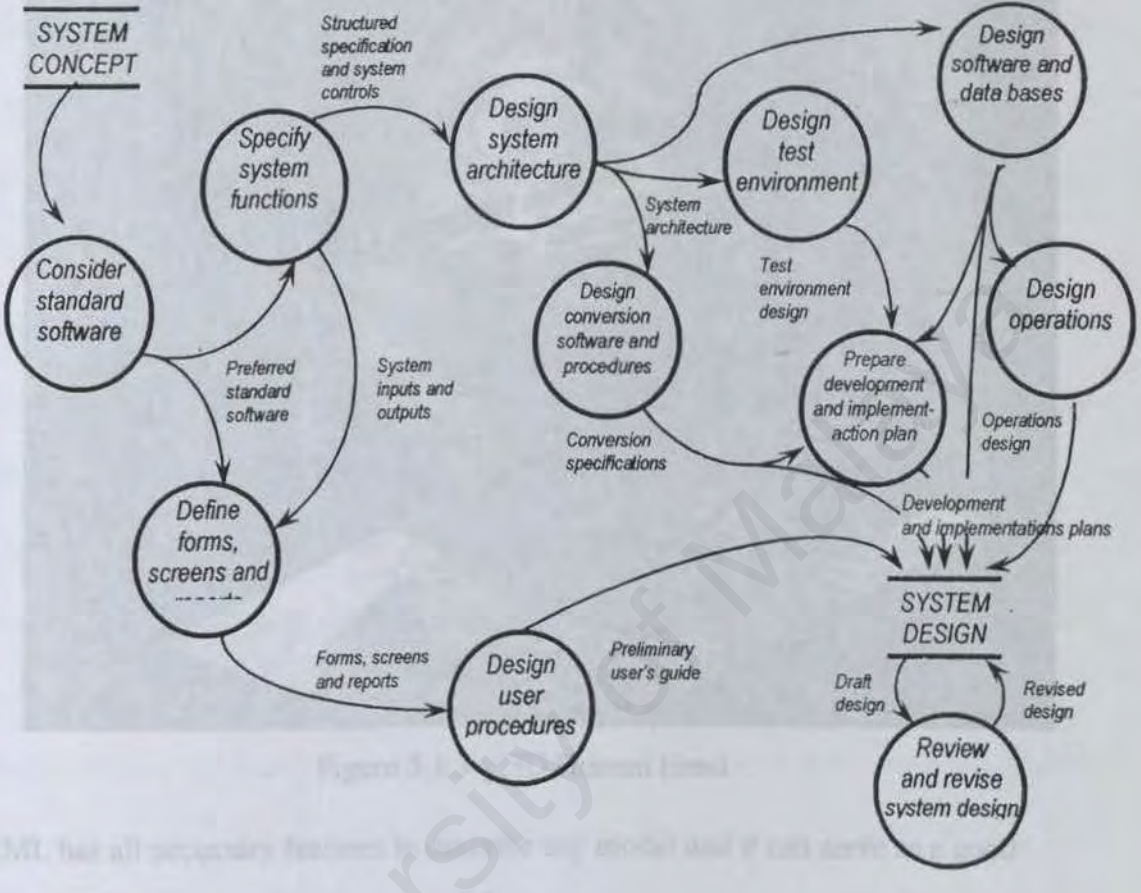


Figure 5.1.2 System design.

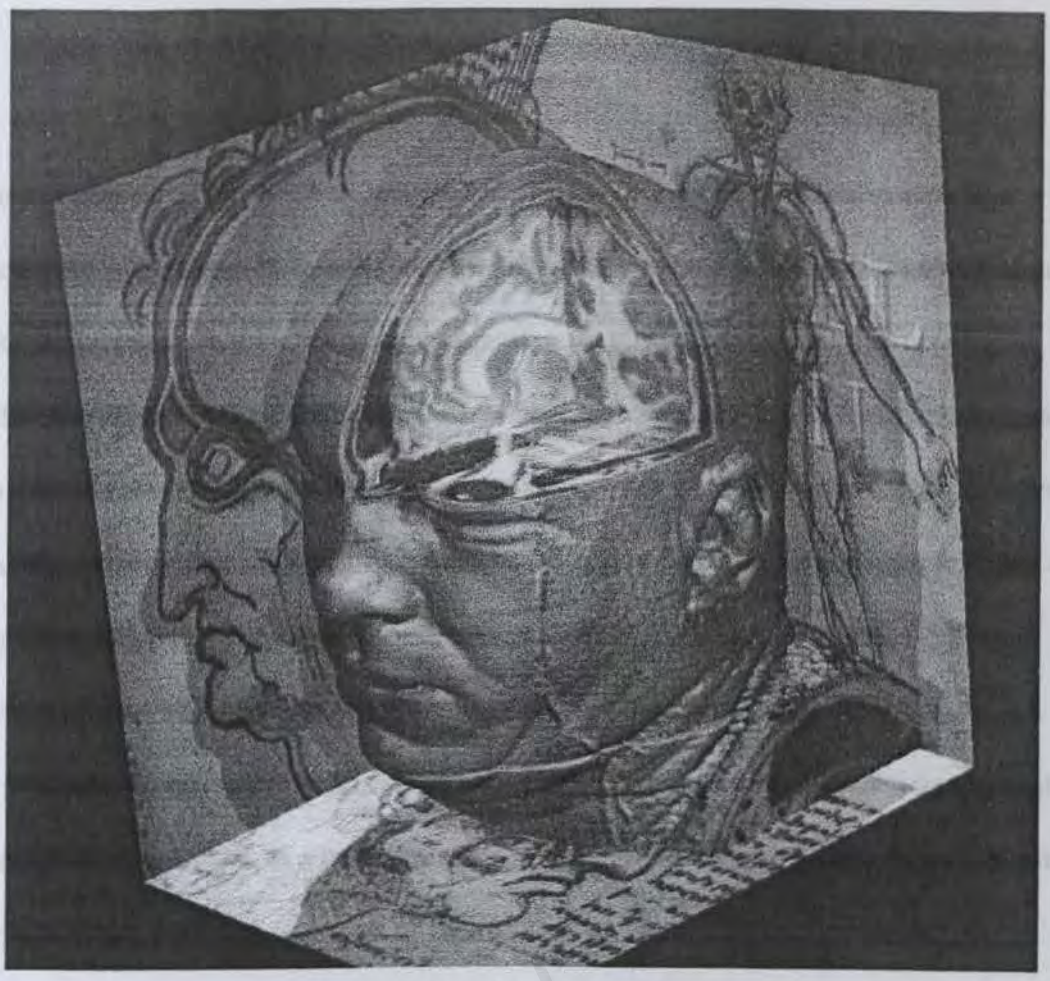


Figure 5.1.3 3D Human Head

VRML has all necessary features to describe any model and it can serve as a good language for model description. VRML has following advantages:

- it supports the definition of structured user data types, therefore we can create model with all properties we need
- we can inspect created model in 3D environment
- we can use the large scale of defined structures and data types

All data are included in one file and it is the main advantages of this model representation. The next advantage is the exact definition of data types and structures. This type of model representation was used in my application.

5.2 Representation of muscles and dividing of model

One of main problems in virtual human systems is an appropriate subdividing of the model and definition and implementation of facial muscles. Main part of virtual human systems depicting human head is implementation of muscles around the mouth. Following pictures contains all muscles, which are important for the lips and the lower jawbone. This subdividing can use some of advantages of the OpenGL structures. The model contains quite large amount of graphic elements, that need to be created once and they must not be recomputed in every moment, when the picture refreshes. It is advantageous to use the OpenGL Display-list.

The Display list is a structure created in memory only once (at the time of list creation). Using of Display-list decrease number of operations. The Display-list has the one major disadvantage contents of the list cannot be modified. Only possibility to change Display-list is to move, rotate and scale list, therefore it is absolutely useless for parts of the model, that change their appearance in different ways than the moving, rotating and scaling. The Display-list can be used for face areas without the facial muscles. It includes areas of temples, some segments of a forehead, the segments around the lower jawbone and the part of the nose. It is necessary to dynamically regenerate the shape changing parts of the model. The areas around mimic muscles are parts of the model that change their shape. In each step of a model refresh positions of vertices and appropriate faces are recomputed and they are redrawn. We divide the model to 6 parts one static part and five dynamical parts. The dynamical parts are: *regio oralis*, *regio*

buccalis, regio mentalis, regio paratideomasseterica and *trigonum submandibulare.* They are regenerated in every step, when they change.

5.3 The Surface Anatomy & Surface Markings

Bones - Various bony surfaces and prominences on the skull can be easily identified by palpation. The external occipital protuberance is situated behind, in the middle line, at the junction of the skin of the neck with that of the head. The superior nuchal line runs lateralward from it on either side, while extending downward from it is the median nuchal crest, situated deeply at the bottom of the nuchal furrow. Above the superior nuchal lines the vault of the cranium is thinly covered with soft structures, so that the form of this part of the head is almost that of the upper portion of the occipital, the parietal, and the frontal bones.

The superior nuchal line can be followed lateralward to the mastoid portion of the temporal bone, from which the mastoid process projects downward and forward behind the ear. The anterior and posterior borders, the apex, and the external surface of this process are all available for superficial examination. The anterior border lies immediately behind the concha, and the apex is on a level with the lobule of the auricular. About 1 cm. below and in front of the apex of the mastoid process, the transverse process of the atlas can be distinguished. In front of the ear the zygomatic arch can be felt throughout its entire length; its posterior end is narrow and is situated a little above the level of the tragus; its anterior end is broad and is continued into the zygomatic bone. The lower border of the arch is more distinct than the upper, which is obscured by the attachment of

the temporal fascia. In front, and behind, the upper border of the arch can be followed into the superior temporal line. In front, this line begins at the zygomatic process of the frontal bone as a curved ridge which runs at first forward and upward on the frontal bone and then curving backward separates the forehead from the temporal fossa. It can then be traced across the parietal bone, where, though less marked, it can generally be recognized. Finally, it curves downward, and forward, and passing above the external acoustic meatus, ends in the posterior root of the zygomatic arch. Near the line of the greatest transverse diameter of the head are the parietal eminences, one on either side of the middle line; further forward, on the forehead, are the frontal eminences, which vary in prominence in different individuals and are frequently unsymmetrical. Below the frontal eminences the superciliary arches, which indicate the position of the frontal sinuses, can be recognized; as a rule they are small in the female and absent in children. In some cases the prominence of the superciliary arches is related to the size of the frontal sinuses, but frequently there is no such relationship. Situated between, and connecting the superciliary ridges, is a smooth, somewhat triangular area, the glabella, below which the nasion (*frontonasal suture*) can be felt as a slight depression at the root of the nose.

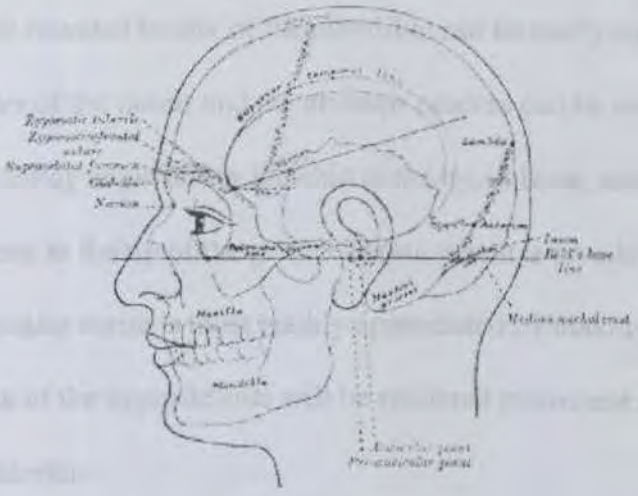


Figure 5.3.1 Side views of head, showing surface relations of bones.

Below the nasion the nasal bones, scantily covered by soft tissues, can be traced to their junction with the nasal cartilages, and on either side of the nasal bone the complete outline of the orbital margin can be made out. At the junction of the medial and intermediate thirds of the supraorbital margin the supraorbital notch, when present, can be felt; close to the medial end of the infraorbital margin is a little tubercle which serves as a guide to the position of the lacrimal sac. Below and lateral to the orbit, on either side, is the zygomatic bone forming the prominence of the cheek; its posterior margin is easily palpable, and on it just above the level of the lateral palpebral commissure is the zygomatic tubercle. A slight depression, about 1 cm. above this tubercle, indicates the position of the zygomaticofrontal suture. Directly below the orbit a considerable part of the anterior surface of the maxilla and the whole of its alveolar process can be palpated. The outline of the mandible can be recognized throughout practically its entire extent; in front of the tragus and below the zygomatic arch is the condyle, and from this the posterior border of the ramus can be followed to the angle; from the angle to the

symphysis the lower rounded border of the mandible can be easily traced; the lower part of the anterior border of the ramus and the alveolar process can be made out without difficulty. In the receding angle below the chin is the hyoid bone, and the finger can be carried along the bone to the tip of the greater cornu, which is on a level with the angle of the mandible: the greater cornu is most readily appreciated by making pressure on one side, when the cornu of the opposite side will be rendered prominent and can be felt distinctly beneath the skin.

Joints and Muscles - The temporomandibular articulation is quite superficial, and is situated below the posterior end of the zygomatic arch, in front of the external acoustic meatus. Its position can be ascertained by defining the condyle of the mandible; when the mouth opens, the condyle advances out of the mandibular fossa on to the articular tubercle, and a depression is felt in the situation of the joint.

The outlines of the muscles of the head and face cannot be traced on the surface except in the case of the Masseter and Temporalis. The muscles of the scalp are so thin that the outline of the bone is perceptible beneath them. Those of the face are small, covered by soft skin, and often by a considerable layer of fat, and their outlines are therefore concealed; they serve, however, to round off and smooth prominent borders, and to fill up what would otherwise be unsightly angular depressions. Thus the Orbicularis oculi rounds off the prominent margin of the orbit, and the Procerus fills in the sharp depression below the glabella. In like manner the labial muscles converging to the lips, and assisted by the superimposed fat, fill up the sunken hollow of the lower part of the face. When in action the facial muscles produce the various expressions, and in addition throw the skin into numerous folds and wrinkles. The Masseter imparts fulness

to the hinder part of the cheek; if firmly contracted, as when the teeth are clenched, its quadrilateral outline is plainly visible; the anterior border forms a prominent vertical ridge, behind which is a considerable fulness especially marked at the lower part of the muscle. The Temporalis is fan-shaped and fills the temporal fossa, substituting for the concavity a somewhat convex swelling, the anterior part of which, on account of the absence of hair on the overlying skin, is more marked than the posterior, and stands out in strong relief when the muscle is in action.

Figure 5.3.2 Anterolateral view of head and neck.

In the neck, the Platysma when contracted throws the skin into oblique ridges parallel with the fasciculi of the muscle. The Sternocleidomastoideus has the most important influence on the surface form of the neck. When the muscle is at rest its anterior border forms an oblique rounded edge ending below in the sharp outline of the sternal head; the posterior border is only distinct for about 2 or 3 cm. above the middle of the clavicle. During contraction, the sternal head stands out as a sharply defined ridge, while the

clavicular head is flatter and less prominent; between the two heads is a slight depression: the fleshy middle portion of the muscle appears as an oblique elevation with a thick, rounded, anterior border, best marked in its lower part. The sternal heads of the two muscles are separated by a V-shaped depression, in which are the Sternohyoideus and Sternothyreoideus. Above the hyoid bone, near the middle line, the anterior belly of the Digastricus produces a slight convexity.

The anterior border of the Trapezius presents as a faint ridge running from the superior nuchal line, downward and forward to the junction of the intermediate and lateral thirds of the clavicle. Between the Sternocleidomastoideus and the Trapezius is the posterior triangle of the neck, the lower part of which appears as a shallow concavity—the supraclavicular fossa. In this fossa, the inferior belly of the Omohyoideus, when in action, presents as a rounded cord-like elevation a little above, and almost parallel to, the clavicle. Arteries - The positions of several of the larger arteries can be ascertained from their pulsations. The subclavian artery can be felt by making pressure downward, backward, and medialward behind the clavicular head of the Sternocleidomastoideus; its transverse cervical branch may be detected parallel to, and about a finger's breadth above, the clavicle. The common and external carotid arteries can be recognized immediately beneath the anterior edge of the Sternocleidomastoideus. The external maxillary artery can be traced over the border of the mandible just in front of the anterior border of the Masseter, then about 1 cm. lateral to the angle of the mouth, and finally as it runs up the side of the nose. The pulsation of the occipital artery can be distinguished about 3 or 4 cm. lateral to the external occipital protuberance; that of the posterior auricular in the groove between the mastoid process and the auricula. The course of the

superficial temporal artery can be readily followed across the posterior end of the zygomatic arch to a point about 3 to 5 cm. above this, where it divides into its frontal and parietal branches; the pulsation of the frontal branch is frequently visible on the side of the forehead. The supraorbital artery can usually be detected immediately above the supraorbital notch or foramen.

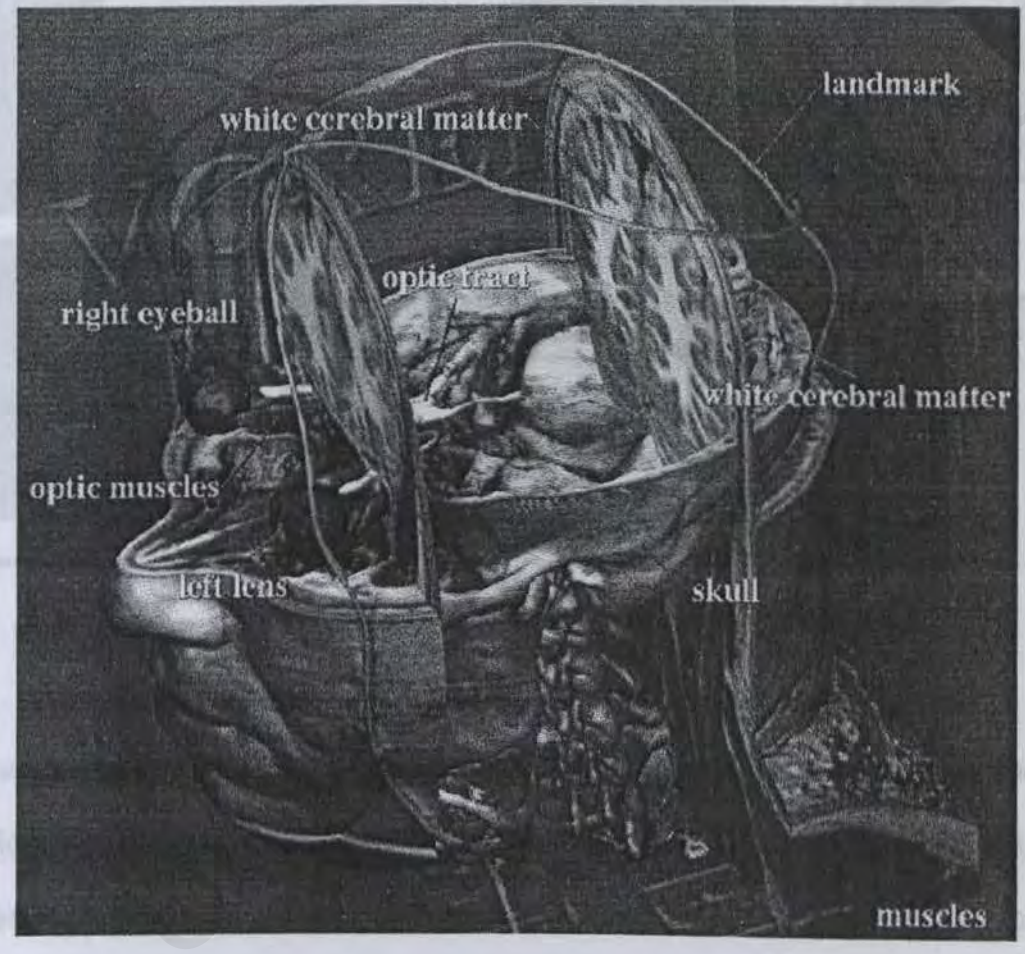


Figure 5.3.3 Inner Head Anatomy

5.4 Screen and Interfaces Design

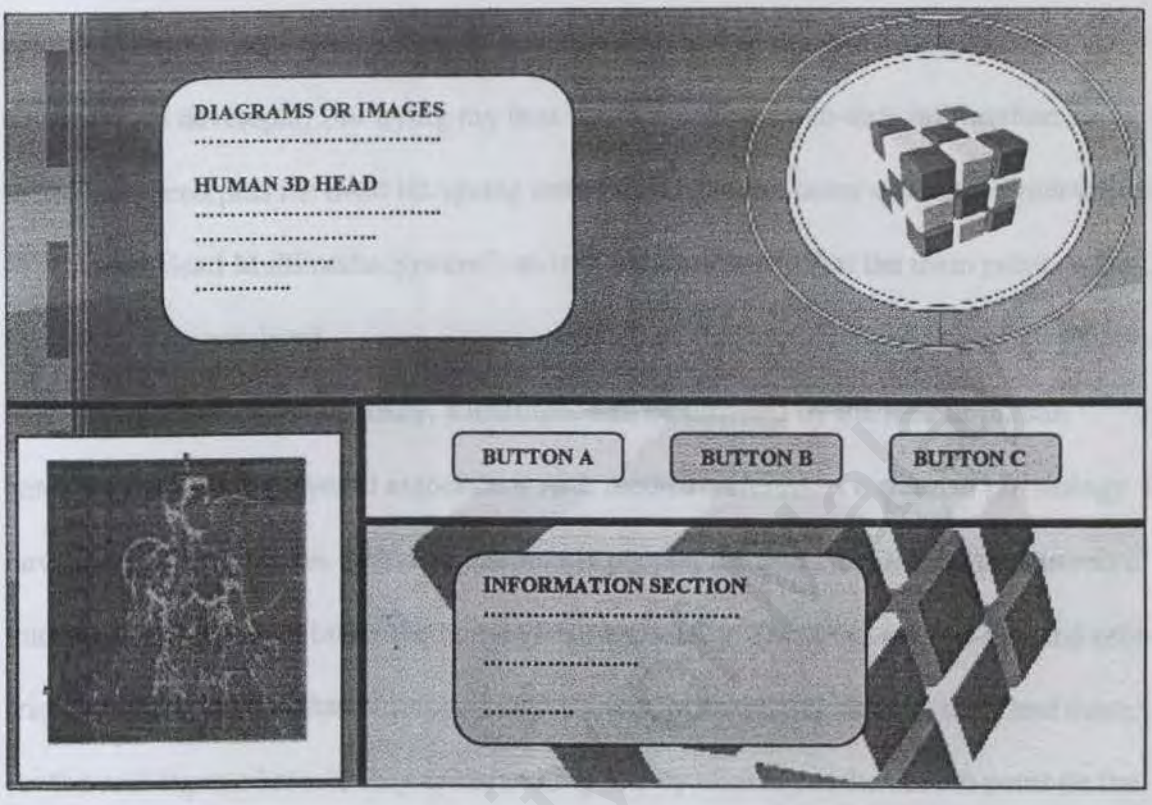


Figure 5.4.1 Screen prototype.

Let me remind that this is just a sample or an example of what the screen and interfaces may look like. This is one of the advantages of prototyping. It's a half-ready product that we can actually inspect for error or dissatisfaction. As for the picture section, users can view the 3d human head model, after that the information about its segment or facial features will display on the information section. The buttons are for navigations.

5.5 Storyboard

What do I know about the user's needs? Honestly, if I'm the user myself, I would like to get the right and trustworthy information when I'm surfing the net. As the system designer and developer, I'm trying my best to get the most up-to-date information about the human head plus the most intriguing interfaces that I can come out with. "Interactive 3D Human Head Multimedia System", as it is seemed clearly that the main point or the key word is human head.

People, the public mostly, sometimes can be terrified by the idea of human anatomy because they would associate it with medical references or human physiology. I have to put a stop on this. I have to encourage people, the users to surf and to gain more knowledge on our own body, the human head especially. This is an easy-to-use and user-friendly multimedia system equipped with sounds, pictures, 3D models, texts and more. As the user input whatever they are searching for, by clicking at the certain point on the 3D human head, the information on that certain segment will display. Within this information section, the user can chose their details on the human head segment. Thus, it will not look too complex for primary and secondary students or the public users.

The main purpose of this project or system is to get as many users possible, by not bore them but to educate them. I'm certain with my ideas on multimedia design and implementation. This project will be the main attraction of the century in the computerized environment.

6.1 INTRODUCTION

The process of assuring that the information system is operational and then allowing users to take over its operation is called system implementation. System implementation is further defined as the construction of a new system and the delivery of that system into production in a day-today operation. It involves coding step that translates a detailed design representation of software into a program language realization. System implementation implements the various components of the system based on the collected requirements, where the design is translated into a machine-readable form.

During implementation, all functionality planned in design phased is checked. It should be able to process the correct data and produce accurate information to end-users. Any problem or malfunction occurred is revised carefully and fixed accordingly. This is important due to the approach for development of my system.

6.2 DEVELOPMENT ENVIRONMENT

The development environment is crucial for the completeness and successfulness of any computer system. Development environment plays a major role in determining the speed of developing the system. During development, the weaknesses will be noticed and improved, while the errors found will be removed.

Using suitable hardware and software will help to speed up system development. Thus, the hardware and software are carefully considered to facilitate the development

of the web-based "Interactive 3D Human Head Multimedia System". System development consists of the used of development tools and methodology chosen, and coding styles. The details are illustrated as below:

6.2.1 Development Tools

6.2.1.1 Hardware Requirements

The following hardware specifications are required to develop the web-based "Interactive 3D Human Head Multimedia System:

- Processor: 800MHz Pentium 3 processor
- Memory: 512MB RAM
- Hard disk: 20GB Hard Disk
- CD-ROM drive: 48x24x48x CD-RW Burner
- Floppy drive: 1.44MB Floppy Drive
- Input device: Mouse, keyboard, scanner, floppy drive and disk drive.
- Output device: Printer, monitor.

6.2.1.2 Software Requirements

The following software specifications have been used to develop the web-based "Interactive 3D Human Had Multimedia System":

- Operating system: Microsoft Windows 98 Plus.

- Web Development Tool:
  - Macromedia Dreamweaver MX
  - Java Script
- Interface Design Tool:
  - Swish 2.0
  - Macromedia Flash MX
- 3D Design Tools:
  - VRML 2.0
  - 3D Studio Max 5.0

6.2.1.3 Methodology

This project is developed using the waterfall with prototype approach. The development of this project consists of five stages, which are requirement, design, coding, testing and operation plus a prototype. The system is design using logical flow and it allows the estimation of the milestones. Each stage must be completed before proceed to the next stage to ensure that the system is built according to the requirements and specifications. Prototyping is such a sub process; a prototype is a partially developed product that enables customers and developers to examine some aspect of the proposed system and decide if it is a suitable or appropriate for the finished product.

6.2.2 System Coding

System coding is a set of instruction written in order to enable the code to be executed and perform the required functionality. A good and well-managed program coding will enhance the readability of the whole program. In addition, it provides an easy understanding to the program flow especially for those programs with high degree of complexity.

6.2.2.1 Coding Approach

Some of the approaches used in the coding development are listed as below:

- Readability

Code document is important to ease the readability of a system. It begins with the selection of identifier (such as variables and labels) names and continues with the composition and organizing the whole program.

- Naming Technique

This is good and meaningful technique of variables, controls and modules that provide easy identification for the program. The naming convention is created with the consistency and standardization in coding.

- Internal Documentation

This provides a clear guideline to developers and readers about the function of a particular source code in the program. Therefore, comments provide the developer with the means of communication with other readers of the source code. The

statement of the module and descriptive comments are embedded within the body of the source code is used to describe the processing function.

- Modularity

The main purpose of modularity is to reduce complexity of system and to facilitate the developer to implement the system by encouraging parallel development of different parts of the system. With the approach of modularity, developer can implement all modules at the same time and does not have to wait for a particular module to complete before going into another module.

VRML coding of which I've implement into my system is a complex coding that I studied for quite some time just to get familiar with it and then started using it to model my 3D models. I have undertaken the complexity of VRML coding by using the code I familiar with and with that I have created the 3D models error free. The coding will be much easier to write and to use when once the developer get the idea and the image of what they are trying to create. VRML coding uses a lot of numbered value that control the size, color, texture, transparency and the movement of a model. By familiarize the codes with the models, the model designing will be hassle free.

6.2.2.2 Coding Style

Coding style is an important component of the source code and it determines the intelligibility of a program. An easy to read source code makes the system easier to be maintained and enhanced in future. Listed below are some of the coding styles used during the coding phase of this project:

- Selection of meaningful identifier names (variables, forms, labels, images and pictures).
- Description and an appropriate comment written in the source code to make it easier for the readers to understand the source code.
- Indentation of codes will increase the readability of the program and for a neater look.
- Meaningful and understandable function and method declarations.
- Keep all complex statement as simple as possible to avoid confusion.

The accuracy of my VRML programming is shown in my web-based Interactive 3D Human Head Multimedia System. As the user browse from page to page, they can see how the 3D models presented. Not only it is 3D models but they can also be rotated or in another way to put it, they can be animated. Just to give the user a little bit of excitement while learning about the human head and to get their interest in the subject.

6.3 PROJECT DEVELOPMENT

After defining all the hardware and software requirements for implementing this project, it's time to concentrate on the development process that involved activities and results of software production. This section will explain how design and system requirements are translated into the form that can be understood by the machine. There are three steps involved, which are data preparation, database connection and coding for each function. But before I explain more on these three steps, I would like

to explain about a few changes that have to be made during this implementation phase.

During the system development stage, I have used new software by the name of Swish 2.0. This software plays the role as the banner design part. Of which, I made two banners for my web site. It can consider as the basic version of Macromedia Flash. Other then that, all the software I stated in Chapter 4 (System Analysis) is used during the system development stage.

6.3.2 Data Preparation

Before developing this project, all of the data involved must be first collected. As for my web-based Interactive 3D Human Head Multimedia System, the main feature of my project is the 3D models and images used to create an educational web site, so the important thing to do is to look for the images, information about the human head, 3D model designs and audio/visual files that are suitable for making the website. There is a lot of Web site that offer free images for personal used so there's no problem for me to find the right images for my project. As for the human head 3D model, I created using 3D Studio Max 5.0 by studying the tutorials that are available on the internet and in the software itself plus a couple of 3D models such as the brain and the eye that I used VRML 2.0. Next, each of the images, 3D models and all the files are saved in two different kinds of folders, which are:

- 'Web Testing' folder – containing all the files that made my web site.
- 'Template' folder – containing templates I created using DreamWeaver MX for the usage of my website.

6.4 SUMMARY

This chapter describes the implementation of the system being developed. It begins with the introduction to the system implementation. System implementation implements the various components of the system based on the collected requirements, where the design is translated into a machine-readable form.

Then, the chapter describes the development environment and project development of the system. The development environment includes of hardware and software requirements specification, methodology chosen, and system coding styles. Whereas the project development consists of the detailed of new software requirements, data preparation, database connection and coding for each function. All of the codes involved are included in Appendices to show the coding environment.

7.1 INTRODUCTION

System testing is a critical element of software quality assurance. It is required to ensure that the system is developed according to its specifications and in line with the users requirements and expectations. Testing is not the first place where faults finding take place but it is focused on finding faults and errors. There are many ways to increase the effectiveness and efficiency of the testing efforts, which will be discussed later in this chapter. Failure of a system can be the results of several reasons:

- The specification may be wrong of have missing requirement and do not state exactly what the customer needs.
- The specification may contain a requirement that is impossible to implement by the given pre described hardware, software and resources.
- The system design phase may contain fault or error that carried forward to the implementation phase.
- The program code may be wrong. Perhaps the algorithm is implemented improperly.

Faults identification is the process of determining what fault causes the failure of the system. The fault correction or removal is the process of making changes to the system so that the fault can be removed.

7.2 TESTING OBJECTIVES

The reason and objectives for performing extensive tests during the design and development of the system are as followed:

- Achieve high quality assurance such as completeness, accuracy, reliability and maintainability of the software program and its documentation.
- Ensure that the system can perform its functions as expected.
- Reduce cost in maintaining the system.
- A method for detection and removal errors.

7.3 TESTING TECHNIQUE

The component of a system will be allowed to manipulate the data, and the output will be observed. Thus, a wide range of inputs and conditions are chosen in order to test that particular component. A test point/test case is a particular choice of input data to be used in testing program.

7.3.1 White Box Testing

White box testing is a testing case design method that uses the control structure of the procedural design to derive test cases. By using white box testing methods, the test cases with the following characteristics can be driven:

- Exercise all logical decision on their true or false side.
- Exercise all loops at their boundaries and within their operational bounds.
- Exercise internal data structure to ensure their validity.
- Guarantee that all independent paths within a module have been exercised at least once.

7.3.2 Black Box Testing

Black box testing focuses on the functionality requirements of the system. It enables the developer to derive sets of inputs condition that will fully exercise all functional requirements for an application. Black box testing was not used as an alternative to white box testing technique rather than this technique is used as a complementary approach that is likely to uncover a different class of errors. Black box testing attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external data access
- Performance access
- Initialization and termination errors.

It also tests the functionality of the system in an ad hoc basis without knowing the logic structure of the code. Input is provided and output is verified manually to check for accuracy.

7.4 TESTING STRATEGY

A strategy to test this system is actually a series of steps that are implemented sequentially. After a program is completely coded, it will be tested under unit testing. Module testing will start when all the programs under a particular module have been completely coded and tested under unit testing. The integration testing is to recover errors associated with interfacing when integrating all the modules.

7.4.1 Unit Testing

Unit testing focuses on verification effort on the smallest component of the system design. Each component is treated as a standalone entity and tested individually to ensure that they operate correctly. The unit test is usually white-box oriented and the step can be conducted in parallel for multiple components.

The test that occurs as part of unit tests is illustrated schematically in Figure 7.0. The module interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that the module operate properly at boundaries established to limit or restrict processing. All independent paths (basis path) are executed at least once. Finally, all error –handling paths are tested.

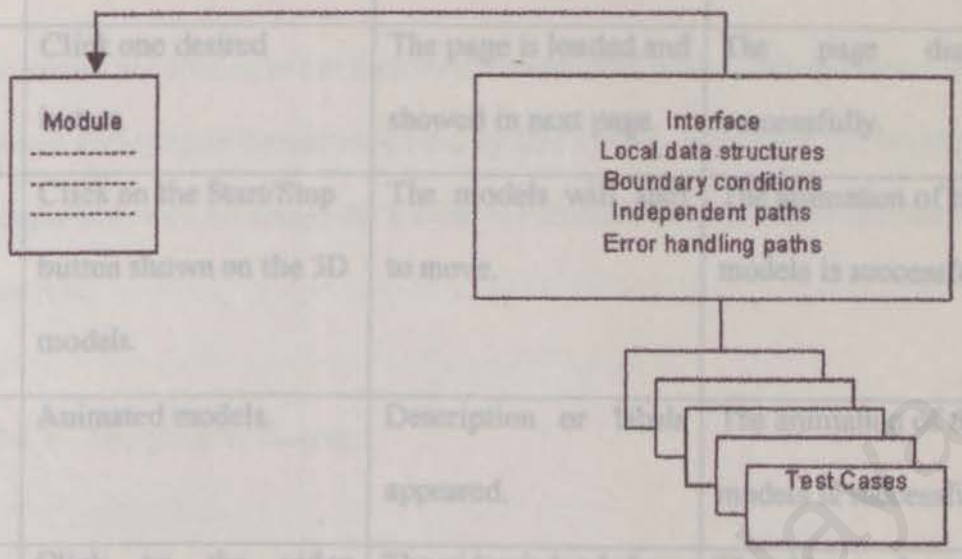


Figure 7.0: Unit Testing

7.4.1.1 Unit Testing Example

Table below shows the test cases for unit testing on the Interactive 3D Human Head Multimedia System.

Step	Test Procedure	Expected Outcome	Test Result Analyzing
1	Click one desired button.	The page is loaded and showed in next page.	The page displayed successfully.
2	Click on the Start/Stop button shown on the 3D models.	The models will start to move.	The animation of the models is successful.
3	Animated models.	Description or labels appeared.	The animation of the models is successful.
4	Click on the video button. Chosen by the user's choice.	The video is loaded and played.	The video is played and no error occurred.
5	Click on the "Search" button.	The search engine page is loaded and ready to search the information typed by the user.	The search engine will successfully search and give the appropriate website submitted by the user.
6	Click the webmaster email.	Microsoft Outlook window will appear and feedback or question (emails) regarding the website will be sent to the webmaster.	The email is successfully sent.

Table 7.0: Unit testing example

7.4.2 Control Object Testing

All the menus are clicked to test their functionality and work stages are tested with the image and audio/visual format supported by this system, which are .gif or .jpg, .jpeg for images and .wav for songs. As for the 3D models, the files representation will be .max or .wrl.

7.4.3 Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. Testing a specific feature together with other newly developed feature is known as integration testing. In other words, when the individual components are working correctly and meet the objectives, these components are combined into a working system.

In this system, a bottom-up approach has been used. Bottom-up integration testing begins construction and testing with modules at the lowest levels of the system and then moving upward to the modules at the higher levels of the system. Regression testing is the re-execution of some subset of tests that already been conducted to ensure that changes have not unintended side effects. It is the activity that helps to ensure that changes (due to testing or other reason) do not introduce unintended behavior or additional errors.

7.4.4 System testing

System testing is a series of different tests designed to fully exercise the software system to uncover its limitations and measure its capabilities. The objective is to test an integrated system and verify that it meets specified requirements. Although each test in this system has a difference, all work to verify that the system elements have been properly integrated and perform allocated functions.

7.5 SUMMARY

This chapter is all about testing. These testing include unit testing, control object testing, integration testing and system testing.

Interactive 3D Human Head Multimedia System has been tested and debugged effectively to achieve the objectives of the system. Through all the testing phases, it is easier to ensure the system's qualities and strengths. Debugging and fixing of the program can be done. The limitations of the system's functionalities can be found and improved.

As a conclusion, testing phase is a very important phase in Interactive 3D Human Head Multimedia System and it must be done repeatedly and carefully to assure good system quality.

8.1 INTRODUCTION

In the process of developing a system, various problems have been identified which some have been solved and some of them are yet to be discovered and overcome. These problems were solved through research and reference books. Besides, a lot of system analysis has been done on technological and programming concepts to grasp the concept of Internet programming.

After all the designing and developing as well as implementing the web-based Interactive 3D Human Head Multimedia System, the end product of the project is brought up for evaluation. The system was evaluated to identify the strengths and the limitations of it. Besides, proposal and recommendations are made for the future enhancements of the system.

8.2 PROBLEMS ENCOUNTERED AND SOLUTIONS

Lack of knowledge in the complex programming of VRML.

Since the first time I got to learn VRML (Virtual Reality Modeling Language) I find that it is a complex programming language to design models and fill colors in it. It require a lot of self-learning and information gathering about this programming language. VRML consists a lot of numbered value that can be change to fit the developer's criteria. Eventually, I have learned a lot from the lecturer and from the Internet that provide step-by-step courses and tutorials. I have end up understanding VRML source code and technique of objects manipulation.

- **Creating 3D Models Using VRML.**

First thing first, right after I understand the VRML coding. It takes a lot of time to really apply the knowledge into the 3D modeling. It's time consuming but the system prevailed. As shown in the system, I have manage to create the 3D models and put some view point into them as the user can watch the 3D models from every view they prefer.

- **Hardware Problems.**

I'm using an AVIDIA TNT graphic card, which doesn't support the setting (Flash) I used for my web site. It require a whole lot better graphic cards such as Voodoo or etc. I manage to swap my system using another computer to preview my system. Other then that, everything goes smoothly as planned.

8.3 SYSTEM STRENGTHS

- **Simple, user-friendly and easy to use**

The interface design of this system mostly was created using Macromedia DreamWeaver MX, Swish 2.0 and Macromedia Flash MX. It is designed to be as user-friendly as this system is relatively easy to learn and use. All the menus used to ease the user explore and try this system by themselves. An action is just a click away and the user just needs minimal knowledge of mouse and keyboard to use this system.

- **A Lot of Interesting Multimedia Functions.**

There's a Video page that I've created to visualize the whole new perspective of see from the title of my system, "Interactive 3D Human Head Multimedia System", it consists of every multimedia functions from plain text, audio, visual, and animation and etc. This video page helps a lot of user to understand the complex mind of a human being. This video page completes with voice description and easy to understand clips.

Besides that, there also background music for the web site. Not for just one page but all the pages in it. So it doesn't make the user get bored. I chose a very tranquilize and relaxed tune to help the user to concentrate on its topic. From my own research in the same related topics, medicine, it is told that music can heals pain, improve concentration and makes a person focused.

- **The Perfect Description of The Human Head.**

From the 3D models it is also provides description of its organs. My target users ranging from primary school students until secondary school students, plus it is beneficial to the public to gain extra knowledge on the human head. When people get to know about my web site, they will love it because all the text and information are easy to understand from any other web site found in the Internet nowadays. So people won't get scare when they want to learn more about the human head despite its medical terms of which I only use the simplest terms in it.

- **Search Engine Available.**

There is a search engine in my web site to help the user to find anything they want from the Net. From my point of view, I think it is a must for a website to provide a search engine.

- **Webmaster E-mail.**

I have put my email on the lower left side on the web site for the user to ask question about my system. And to learn more about myself. By adding this components in my system I got to know a lot of people around the world and exchanging views on web site designing, the anatomy of the human head and everything that is interesting.

- **Others.**

Besides everything that I have stated above, there's also a fast fact page. In this page it provide a lot of interesting facts about the human head. The user can use this information to strengthen their knowledge and quiz themselves with their friends.
A calendar and a clock is located on the lowest left side of the web site for the user to keep in touch with time.

8.4 SYSTEM CONSTRAINTS

- **Wizard.**

Referring to the title of my project, which is Interactive 3D Human head Multimedia System, this system should offer a wizard to help user to explore the website but due to my lack of knowledge, experience and guidance, I was only able to accomplish the template part. In addition, after conducting a few more research about this problem, I have found that it was impossible to implement the function in Web application. Furthermore, so far the function can only be seen in stand-alone software, which is not Web based application, normally such as in CD-ROM type.

- **No search keyword function.**

No search keyword function in this system. This is also due my lack of knowledge, experience and guidance. I think function is necessary because it can help the user to explore even more through web pages to find their desire topics.

8.5 FUTURE ENHANCEMENTS

System development is a dynamic process and changes must be expected. Due to limited resources, especially time, had caused me to miss or overlooks certain aspect of the system. However, after the development system has been completed and valuable advices and suggestions from my project supervisor and moderator, I have

After explaining earlier in system constraint section about this feature, I think with enhanced tools in the future, this function is no longer impossible to implement.

- **Two versions of the human head.**

I think there should be 2 versions of the 3D human. One is for the public to surf to and the other one is for them with medical background of which even more details models and description is provided.

8.6 KNOWLEDGE AND EXPERIENCE GAINED

Towards the accomplishment of Interactive 3D Human Head Multimedia System, from the beginning to the end of the development and final documentation, a number of problems and difficulties are encountered. However, the solutions to these problems and difficulties have brought numerous valuable knowledge and experience. Following are the benefits and knowledge gained from this project:

- **The importance of all phase in SDLC**

System analysis is an important phase in the System Development Life Cycle (SDLC). This phase captures user requirements and the goal of the system. If this phase is mistakenly defined, it will cause faulty to the system development and later progress. With a complete and thorough system analysis, the system that is developed will fulfill all the requirements and achieve it goals.

System testing is also an important and critical phase in SDLC. There is no application that is free of error in this world. However, with the procedures in the system-testing phase, errors and faults in the system can be minimized.

- **Development tools knowledge**

This project is developed using VRML (Virtual Reality Modeling Language) and it is so useful for me to learn one of many software out there in the multimedia world today. Not just its provide me with skill but the knowledge gained was priceless.
Also, with the use of other tools such as Macromedia Dreamweaver MX, Macromedia Flash MX, Swish 2.0 and 3D Studio Max, my knowledge in developing Web site and 3D modeling will never be limited anymore.

8.7 SUMMARY

Evaluation of system is indeed to ensure its objectives and intended functions have been achieved. This chapter covers all the aspects of the evaluating application software. The successful development of the system at the present is the first step towards the future expansion of the system. The problem encountered and experience gained during the development phases should be helpful in future efforts.

Besides, this chapter also summarizes the system strengths, system constraints and future enhancements that can be added. The future enhancements will equip the system towards more capabilities of doing its daily operations and activities.

Conclusion

Interactive 3D Human Head Multimedia System is one of many educational human head anatomy Web sites available in the World Wide Web nowadays. It offers the user a glimpse of the inner and out part of the human head plus the descriptions as well. It is an educational web site that is suitable for all type of users, regardless of age, race or status. However, the system will become more complete and capable of performing more tasks when the enhancement and the new features are added on in the near future.

In the process of developing this system, invaluable insight was gained into complexities and intricacies of programming. The application of Software Engineering principles, fundamentals and additional knowledge in programming languages, skills coding writing and others all added up to contribute to the success of developing this system. Adhere to development schedule is crucial in determining that a system will be completed in time. The experience gathered in this project will definitely provide a solid foundation in the system development in the future.

With target goals and objectives in mind even before the development takes place, makes the development process more systematic. Sometimes, conflicts in real world situation and programming tools capabilities make the programming difficult. However, as an overall review, this project has achieved and fulfilled the objectives though its not meets the requirements determined during the analysis phase entirely.

Bibliography

1. Adelstein, B., Johnston, E., Ellis, S., "A testbed for Characterizing Dynamic Response of Virtual Environment Spatial Sensors," UIST'92, 15-22.
2. Arvo, J., "Random Rotation Matrices." In Graphics Gems II, Arvo, J., ed., Academic Press, pp. 355-356.
3. Ascension technology Corporation, PO Box 527, Burlington, VT 05402. (802) 655-7879.
4. Bolt, R., "Put-That-There: Voice and Gesture at the Graphics Interface," SIGGRAPH ' 80, 262-70.
5. SenAble Devices, Inc., he Phantom Haptic Interfaces. URL: http://www.tiac.net/users/sensable/